The IoT Hacker’s Handbook

The IoT Hacker’s Handbook

Um guia prático para Hacking da internet das coisas

Introdução

Os dez capítulos deste livro abrangem uma série de tópicos, desde exploração de hardware e firmware embarcado até comunicação por rádio, incluindo exploração de BLE e ZigBee.

Para mim, escrever este livro foi uma jornada emocionante e cheia de aventura, compartilhando minhas experiências e as várias coisas que aprendi em minha carreira profissional e colocando tudo nesses dez capítulos.

Espero que você possa aproveitar ao máximo este livro e eu altamente te encorajo a pegar todos os conjuntos de habilidades aprendidos aqui e aplicá-los a problemas do mundo real, ajudando a tornar o ecossistema da Internet das Coisas (IoT) mais seguro. São as contribuições individuais que nos ajudarão a criar um mundo mais seguro, e você, que está lendo este livro, pode fazer parte disso.

Ninguém é perfeito, e este livro certamente terá um ou dois pequenos erros. Se você encontrar algum desses erros, por favor, me avise e ficarei feliz para corrigi-los em futuras edições do Manual do Hacker da IoT.

Eu também dou aulas de três e cinco dias sobre exploração de IoT ofensiva, que eu recomendo que você participe para obter experiência prática com tudo o que é abordado no livro. Para mais informações sobre o treinamento online e as aulas ao vivo, acesse attifystore.com.

A última e mais importante parte é a comunidade! Para você, leitor, eu quero que esteja disposto o suficiente para compartilhar seu conhecimento com seus colegas ou mesmo com alguém que é novo nesta área. É assim que nós, como comunidade, cresceremos.

É tudo da minha parte. Mais uma vez, obrigado por ler o Manual do Hacker da IoT e desejo a todos o melhor em seus empreendimentos de exploração de IoT.

Aditya Gupta (@adi1391)

Fundador e Chefe Hacker,

Attify

Agradecimentos

Esse livro nunca teria sido terminado sem o meu time incrível da Attify, que trabalharam dia e noite para ter certeza que produzimos um conteúdo de qualidade como uma equipe.

1.Internet das coisas: Um primer

No mundo da tecnologia da comunicação, dois dos eventos que possuem significância especial são a invenção da ARPANET, uma rede de computadores que permite a troca de dados entre computadores mesmo quando estão geograficamente separados, e a ascensão da Internet das Coisas (IoT). Este último, no entanto, foi um processo evolutivo ao invés de um único evento.

As primeiras implementações do conceito de IoT ocorreram quando dois estudantes da Universidade Carnegie Mellon encontraram uma maneira de monitorar o número de latas restantes em uma máquina de venda automática permitido que dispositivos se comuniquem com o mundo externo. Eles fizeram isso adicionando um fotossensor ao dispositivo que contaria toda vez que uma lata saía da máquina, e assim, o número de latas restantes era calculado. Hoje em dia, os dispositivos IoT são capazes de monitorar sua frequência cardíaca e até mesmo controlá-la, se necessário, em caso de um evento adverso. Além disso, alguns dispositivos IoT agora podem servir como fonte de evidência durante julgamentos em tribunais, como visto no final de 2015, quando os dados do FitBit de uma mulher foram usados em um julgamento de assassinato. Outros incidentes incluem o uso de dados de marcapasso e gravações do Amazon Echo em vários julgamentos. A jornada dos dispositivos IoT de um dormitório universitário para dentro de seres humanos é fascinante, para dizer o mínimo. Kevin Aston, quando mencionou o termo Internet das Coisas pela primeira vez, provavelmente não imaginaria que esses dispositivos logo ultrapassariam a população humana total em número. Aston mencionou o termo em referência à tecnologia de identificação por radiofrequência (RFID), que estava sendo usada para conectar dispositivos entre si. A definição de IoT mudou desde então, com diferentes organizações dando seu próprio significado ao termo. A Qualcomm e a Cisco criaram o termo Internet of Everything (IoE), que alguns acreditam ser para uma agenda de marketing. O termo, de acordo com eles, significa estender o conceito de IoT de ser limitado à comunicação máquina-a-máquina para máquinas se comunicando com máquinas e com o mundo físico.

O primeiro vislumbre da IoT atual foi visto em junho de 2000, quando a primeira geladeira conectada à Internet, a Internet Digital DIOS, foi revelada pela LG. A geladeira continha uma tela TFT-LCD de alta qualidade com uma série de funcionalidades, incluindo exibir a temperatura dentro da geladeira, fornecer pontuações de frescura dos itens armazenados e usar a funcionalidade da webcam para rastrear os itens armazenados. O dispositivo inicial que provavelmente chamou mais atenção da mídia e dos consumidores foi o Nest Learning Thermostat em outubro de 2011. Este dispositivo era capaz de aprender a programação do usuário para ajustar diferentes temperaturas desejadas em diferentes horas do dia. A aquisição desta empresa de termostato IoT pelo Google por US$ 3,2 bilhões foi o evento que conscientizou o mundo sobre a revolução tecnológica que se aproximava.

Logo, havia centenas de novas startups tentando interconectar todos os diferentes aspectos do mundo físico a dispositivos e grandes organizações criando equipes internas especializadas para criar sua própria linha de dispositivos IoT para serem lançados no mercado o mais rápido possível. Essa corrida para criar novos dispositivos chamados de inteligentes nos leva ao presente, onde podemos interagir com nossas smart TVs em casa enquanto tomamos uma xícara de café preparada por uma máquina de café controlada pela Internet e controlamos as luzes pela música tocando em seu assistente

inteligente. A IoT, no entanto, não se limita apenas ao nosso espaço físico. Ele também possui inúmeras aplicações em empresas, lojas de varejo, saúde, indústria, redes de energia e até mesmo em pesquisas científicas avançadas.

Os formuladores de políticas do mundo digital lutaram contra o ritmo acelerado do surgimento de dispositivos IoT e não conseguiram criar controles de qualidade e regulamentações de segurança rígidos. Isso só mudou recentemente, quando organizações como a GSMA criaram diretrizes de segurança e privacidade para dispositivos IoT, e a Comissão Federal de Comércio (FTC) Foram definidas etapas a serem seguidas para garantir a segurança. No entanto, a demora levou à adoção generalizada de dispositivos de IoT em todos os setores, e também permitiu que os desenvolvedores ignorassem as considerações de segurança no que diz respeito a esses dispositivos. Não foi até o efeito generalizado do botnet Mirai que as deficiências de segurança desses dispositivos se tornaram aparentes Dispositivos seriam vulneráveis. Mirai era uma botnet que atacava dispositivos de Internet das Coisas (IoT), principalmente câmeras conectadas à internet, verificando o status das portas 23 e 2323 e forçando a autenticação por força bruta usando credenciais comuns. Sem surpresa, muitas das câmeras IP expostas à internet tinham acesso telnet disponível com um nome de usuário e senha extremamente comuns, o que era fácil de encontrar. A mesma botnet também foi usada posteriormente para assumir o controle da infraestrutura de internet da Libéria, bem como da DYN, o que levou a um ataque a vários sites populares, incluindo GitHub, Twitter, Reddit e Netflix.

Nos últimos anos, embora a segurança desses dispositivos tenha melhorado lentamente, ainda não atingiu um ponto onde eles possam ser considerados extremamente seguros de usar. Em novembro de 2016, quatro pesquisadores de segurança - Eyal Ronen, Colin O'Flynn, Adi Shamir e Achi-Or Weingarten - criaram um interessante worm de prova de conceito (PoC) que atacava usando drones e tomava o controle das luzes inteligentes Philips Hue de um prédio comercial. Mesmo que o ataque tenha sido apenas um PoC, não é exagero pensar que veríamos ransomware para dispositivos inteligentes semelhante ao WannaCry, pedindo dinheiro para abrir a fechadura da porta ou ligar um marcapasso. Quase todos os dispositivos inteligentes comprovadamente possuem problemas críticos de segurança e privacidade, incluindo sistemas de automação residencial inteligente, dispositivos vestíveis, monitores de bebês e até mesmo brinquedos sexuais pessoais. Considerando a quantidade de dados íntimos que esses dispositivos coletam, é assustador ver o quanto estamos expostos a ataques cibernéticos.

O aumento de incidentes de segurança em dispositivos IoT também levou a uma demanda crescente por profissionais de segurança IoT, atuando tanto na construção quanto na quebra desses sistemas. Isso permite que as organizações garantam que seus dispositivos estejam protegidos das vulnerabilidades que invasores mal-intencionados podem usar para comprometer seus sistemas. Além disso, várias empresas começaram a oferecer recompensas por bugs (bug bounties) para incentivar pesquisadores a avaliar a segurança de seus dispositivos IoT, algumas até mesmo enviando dispositivos físicos gratuitos para esses pesquisadores. Nos próximos anos, essa tendência deve crescer e, com o aumento das soluções de IoT no mercado, haverá uma demanda maior por profissionais especializados em segurança IoT no mercado de trabalho.

Problemas anteriores de segurança de IoT

A melhor forma de aprender sobre segurança desses dispositivos é olhar o que aconteceu no passado. Ao aprender sobre os erros de segurança cometidos por outros desenvolvedores de produtos no passado, podemos entender quais tipos de problemas de segurança devemos esperar no produto que estamos avaliando. Mesmo que alguns termos pareçam desconhecidos aqui, nós os discutiremos em detalhes nos próximos capítulos.

Termostato Nest

O artigo “Smart Nest Thermostat: Um Espião Inteligente em Sua Casa”, de Grant Hernandez, Orlando Arias, Daniel Buentello e Yier Jin, menciona algumas das deficiências de segurança do Google Nest que permitiam a instalação de um novo firmware malicioso no dispositivo. Isso era feito pressionando o botão do Nest por cerca de 10 segundos para acionar a reinicialização global. Nesse estágio, o dispositivo poderia ser configurado para procurar firmware em mídia USB comunicando-se com o pino sys_boot5. Na unidade USB, estava presente um firmware malicioso, que o dispositivo então utilizava durante a inicialização.

Jason Doyle identificou outra vulnerabilidade nos produtos Nest que envolvia o envio de um valor personalizado nos detalhes do identificador de conjunto de serviços Wi-Fi (SSID) via Bluetooth para o dispositivo alvo, o que então travava o dispositivo e acabava por reiniciá-lo. Isso também permitiria a um ladrão invadir a casa durante a reinicialização do dispositivo (cerca de 90 segundos) sem ser flagrado pela câmera de segurança Nest.

Casa inteligente Philips

Dispositivos domésticos da Philips sofreram de uma série de problemas de segurança em toda a linha de produtos. Isso inclui o popular worm Philips Hue, criado como uma prova de conceito (PoC) pelos pesquisadores de segurança Eyal Ronen, Adi Shamir, Achi-Or Weingarten e Colin O'Flynn. No PoC, eles demonstraram como as chaves de criptografia simétricas embutidas usadas pelos dispositivos Philips poderiam ser exploradas para obter controle sobre os dispositivos alvo através do ZigBee. Também incluía a infecção automática de lâmpadas Philips Hue colocadas próximas umas das outras.

Em agosto de 2013, Nitesh Dhanjani, um pesquisador de segurança, também surgiu com uma nova técnica para causar blecautes permanentes usando uma técnica de ataque por replay para obter controle dos dispositivos Philips Hue. Ele descobriu essa vulnerabilidade depois de perceber que os dispositivos inteligentes Philips Hue consideravam apenas o MD5 do endereço de controle de acesso de mídia (MAC) como o único parâmetro para validar a autenticidade de uma mensagem. Como o invasor pode facilmente aprender o endereço MAC do host legítimo, ele pode criar um pacote malicioso indicando que veio do host genuíno e com os pacotes de dados com o comando para desligar a lâmpada. Fazer isso continuamente permitiria ao invasor causar um blecaute permanente, sem que o usuário tivesse outra opção a não ser substituir a lâmpada.

O Philips Hue (e muitos outros dispositivos inteligentes hoje) usa uma tecnologia chamada ZigBee para trocar dados entre os dispositivos, minimizando o consumo de recursos. O mesmo ataque que era possível no dispositivo usando pacotes Wi-Fi também seria aplicável ao ZigBee. No caso do ZigBee, tudo o que um invasor precisa fazer é simplesmente capturar os pacotes ZigBee para uma solicitação legítima e simplesmente reproduzi-los para realizar a mesma ação em um momento posterior e assumir o controle do dispositivo. Veremos também como capturar e reproduzir pacotes ZigBee no Capítulo 10.

Lâmpada inteligente Lifx

Dispositivos de casa inteligente têm sido um dos alvos de pesquisa mais populares entre a comunidade de segurança. Outro exemplo inicial ocorreu quando Alex Chapman, um pesquisador de segurança da empresa Context, descobriu sérias vulnerabilidades de segurança na lâmpada inteligente Lifx, tornando possível para invasores injetarem pacotes maliciosos na rede, obter credenciais WiFi descriptografadas e assumir o controle das lâmpadas inteligentes sem qualquer autenticação.

Os dispositivos, neste caso, estavam se comunicando usando 6LoWPAN, que é outro protocolo de comunicação de rede (assim como ZigBee) construído sobre o 802.15.4. Para farejar os pacotes 6LoWPAN, Chapman usou um Atmel RzRaven flashed com a imagem de firmware Contiki 6LoWPAN, permitindo que ele examinasse o tráfego entre os dispositivos. A maior parte da troca de dados confidenciais ocorrendo nessa rede era criptografada, o que fazia com que o produto parecesse bastante seguro.

Uma das coisas mais importantes durante o teste de penetração de IoT é a capacidade de olhar para o produto como um todo, em vez de apenas olhar para um único componente para identificar os problemas de segurança. Isso significa que para descobrir como os pacotes estão sendo criptografados na comunicação via rádio, a resposta provavelmente está no firmware. Uma das técnicas para obter o binário do firmware de um dispositivo é despejá-lo por meio de técnicas de exploração de hardware, como JTAG, que abordaremos no Capítulo 6. No caso das lâmpadas Lifx, o JTAG deu acesso ao firmware Lifx, que, quando revertido, levou à identificação do tipo de criptografia, que neste caso era o Advanced Encryption Standard (AES), a chave de criptografia, o vetor de inicialização e o modo de bloco usado para criptografia. Como essas informações seriam as mesmas para todas as lâmpadas inteligentes Lifx, um invasor poderia assumir o controle de qualquer lâmpada e invadir o Wi-Fi porque o dispositivo também estava comunicando as credenciais de Wi-Fi pela rede de rádio, que agora poderia ser descriptografada.

O Hack do “Jeep”

O Hack do Jeep é provavelmente o hack de IoT mais popular de todos os tempos. Dois pesquisadores de segurança, Dr. Charlie Miller e Chris Valasek, demonstraram em 2015 como poderiam assumir e controlar remotamente um Jeep usando vulnerabilidades no sistema Uconnect da Chrysler, resultando no recall de 1,4 milhão de veículos pela Chrysler.

O hack completo se aproveitou de muitas vulnerabilidades diferentes, incluindo extensos esforços na engenharia reversa de vários binários e protocolos individuais.

Uma das primeiras vulnerabilidades que tornou o ataque possível foi o software Uconnect, que permitia a qualquer pessoa se conectar remotamente a ele por meio de uma conexão celular. A porta 6667 estava acessível com autenticação anônima habilitada e foi encontrada executando D-Bus sobre IP, que é usado para comunicação entre processos. Depois de interagir com o D-Bus e obter uma lista de serviços disponíveis, um dos serviços com o nome NavTrailService tinha um método de execução que permitia aos pesquisadores executar código arbitrário no dispositivo. A Figura 1-1 mostra o código de exploração usado para abrir um shell root remoto na unidade principal.

http://illmatics.com/Remote%20Car%20Hacking.pdf

Uma vez que a execução arbitrária de comandos foi obtida, tornou-se possível realizar um movimento lateral e enviar mensagens CAN assumindo o controle dos vários elementos do veículo, como volante, freios, faróis e assim por diante.

Belkin Wemo

Belkin Wemo é uma linha de produtos que oferece automação residencial completa aos consumidores.

Belkin Wemo é um caso interessante em que os desenvolvedores tomaram precauções para impedir que invasores instalassem firmware malicioso no dispositivo. As atualizações de firmware para Belkin Wemo, no entanto, aconteciam por um canal não criptografado, o que permitia aos invasores modificar o pacote binário do firmware durante a atualização. Como medida de proteção, o Belkin Wemo utilizava um mecanismo de distribuição de firmware criptografado baseado em GNU Privacy Guard (GPG) para que o dispositivo não aceitasse pacotes de firmware malicioso injetados por um invasor. Essa proteção de segurança foi superada com extrema facilidade porque o dispositivo estava distribuindo a chave de assinatura do firmware junto com o firmware durante o processo de atualização, tudo em um canal não criptografado. Um invasor poderia, portanto, modificar facilmente o pacote, bem como assiná-lo com a chave de assinatura correta, e o dispositivo aceitaria alegremente este firmware. Essa vulnerabilidade foi descoberta por Mike Davis da IOActive no início de 2014 e recebeu uma pontuação (CVSS) de 10.0 pela criticidade da vulnerabilidade.

Mais tarde, descobriu-se que o Belkin Wemo tinha vários outros problemas de segurança, incluindo bugs como injeção de SQL e modificação do nome do dispositivo para executar JavaScript arbitrário no smartphone Android do usuário, entre outros. Pesquisas adicionais foram realizadas no Belkin Wemo pelo grupo FireEye (consulte https://www.fireeye.com/blog/threatresearch/2016/08/embedded_hardwareha.html), que envolveu a obtenção de acesso ao firmware e console de depuração usando técnicas de hardware Universal Asynchronous Receiver Transmitter (UART) e Serial Peripheral Interface (SPI). Isso também os levou a identificar que, por meio do acesso ao hardware, alguém pode modificar facilmente os argumentos do bootloader, tornando inútil a verificação de assinatura do firmware do dispositivo.

Bomba de insulina

Um pesquisador de segurança chamado Jay Radcliffe, trabalhando para a Rapid7, identificou que alguns dispositivos médicos, especificamente bombas de insulina, poderiam estar sofrendo de uma vulnerabilidade a ataques baseados em replay. Radcliffe, ele próprio diabético tipo 1, decidiu pesquisar uma das bombas de insulina mais populares do mercado, o sistema de bomba de insulina OneTouch Ping da Animas, subsidiária da Johnson & Johnson. Durante a análise, ele descobriu que a bomba de insulina usava mensagens de texto claro para se comunicar, o que tornava extremamente simples para qualquer pessoa capturar a comunicação, modificar a dose de insulina a ser administrada e retransmitir o pacote. Quando ele testou o ataque na bomba de insulina OneTouch Ping, funcionou perfeitamente, sem nenhuma maneira de saber a quantidade de insulina que estava sendo administrada durante o ataque.

A vulnerabilidade foi corrigida pelo fornecedor, Animas, em cinco meses, o que mostra que pelo menos algumas empresas levam os relatórios de segurança a sério e tomam medidas para manter os clientes seguros.

Fechaduras Inteligentes

Um pesquisador de segurança com o apelido Jmaxx embarcou em um desafio para encontrar pontos fracos de segurança na fechadura inteligente August, considerada uma das fechaduras inteligentes mais populares e seguras, usada tanto por consumidores em suas casas quanto por anfitriões do Airbnb para permitir que os hóspedes façam o check-in quando for conveniente.

Algumas das vulnerabilidades que ele descobriu incluíam a capacidade dos hóspedes de se tornarem administradores modificando um valor no tráfego de rede de usuário para superusuário, firmware não assinado, funcionalidade do aplicativo para ignorar fixação de Secure Sockets Layer (SSL) (habilitando modo de debug) e muito mais.

No mesmo evento, os pesquisadores de segurança Anthony Rose e Ben Ramsey da empresa de segurança Merculite fizeram outra apresentação intitulada “Arrombando fechaduras Bluetooth Low Energy a 400 metros de distância”, na qual revelaram vulnerabilidades em uma série de produtos de fechaduras inteligentes, incluindo Quicklock Padlock, iBluLock Padlock, Plantraco Phantomlock, Ceomate Bluetooth Smart Doorlock, Elecycle EL797 e EL797G Smart Padlock, Vians Bluetooth Smart DoorlockOkidokey Smart Doorlock, Poly-Control Danalock Doorlock, Mesh Motion Bitlock Padlock e Lagute Sciener Smart Doorlock.

As vulnerabilidades descobertas por Rose e Ramsey eram de vários tipos, incluindo transmissão da senha em texto puro, suscetibilidade a ataques baseados em replay, engenharia reversa de aplicativos móveis para identificar informações confidenciais, fuzzing e spoofing de dispositivo. Por exemplo, durante o processo de redefinição de senha, o Quicklock Padlock envia um pacote Bluetooth Low Energy (BLE) contendo o opcode, a senha antiga e a nova senha. Como até mesmo a autenticação normal, acontece através de comunicação de texto puro, um invasor pode então usar a senha para definir uma nova senha para a fechadura da porta, tornando o dispositivo inútil para o proprietário original. A única maneira de reiniciá-lo seria remover a bateria do dispositivo após abrir o compartimento. Em outro dispositivo, o Danalock Doorlock, é possível fazer engenharia reversa do aplicativo móvel para identificar o método de criptografia e encontrar a chave de criptografia codificada ("thisisthesecret") usada.

Hackeando armas e rifles inteligentes

Além dos dispositivos e eletrodomésticos inteligentes típicos, os rifles também estão ficando inteligentes. A TrackingPoint, fabricante de tecnologia para rifles inteligentes, oferece um aplicativo móvel para visualizar e ajustar a mira do tiro. Este aplicativo foi considerado vulnerável a alguns problemas de segurança. Runa Sandvik e Michael Auger identificaram vulnerabilidades no rifle inteligente que lhes permitiram acessar interfaces de programação de aplicativos de administração (APIs) após obter acesso ao dispositivo via UART. Explorando o aplicativo móvel, um ataque baseado em rede permitiria a um invasor alterar vários parâmetros, como velocidade e direção do vento, peso da bala e outros parâmetros necessários para o disparo. Quando esses parâmetros são modificados, o atirador não saberia que essas alterações foram feitas.

Outro caso ocorreu quando um pesquisador de segurança conhecido como Plore conseguiu burlar algumas das restrições de segurança aplicadas pela IP1, uma arma inteligente da Armatix. A arma inteligente exigia que o atirador usasse um relógio especial fornecido pela IP1 para disparar a arma. Para contornar as restrições de segurança, Plore inicialmente realizou uma análise de sinal de rádio e encontrou a frequência exata que a arma usa para se comunicar. Mais tarde, ele percebeu que usando alguns ímãs, o pino de metal que trava o pino de disparo poderia ser manipulado, permitindo ao atirador disparar a bala. Mesmo que o uso de ímãs não seja um ataque de alta tecnologia que você possa pensar ser necessário para explorar dispositivos IoT, é um ótimo exemplo de como pensar fora da caixa pode ajudar a identificar vulnerabilidades.

Estas vulnerabilidades servem como exemplos para ajudá-lo a entender vários tipos de vulnerabilidades normalmente encontradas em dispositivos IoT. Mais tarde, abordaremos vários componentes de dispositivos IoT, incluindo técnicas para exploração de hardware, rádio, firmware e software. Você aprenderá mais sobre como usar algumas dessas técnicas nos dispositivos IoT que você está pesquisando ou realizando um teste de penetração.

Fragmentação da Internet das coisas

Como a IoT é um campo enorme, com toda empresa querendo sua fatia do bolo, você frequentemente encontrará vários protocolos e frameworks que podem ajudar os desenvolvedores a levar seus produtos ao mercado mais rapidamente.

Frameworks de IoT são várias ofertas disponíveis que ajudam os desenvolvedores de IoT a acelerar o processo de desenvolvimento de uma solução de dispositivo IoT, aproveitando a base de código e bibliotecas existentes oferecidas, reduzindo o tempo de lançamento do produto no mercado. Embora isso torne as coisas significativamente mais fáceis para desenvolvedores e empresas, o outro lado, que muitas vezes é negligenciado, é a segurança desses frameworks. Na verdade, com base em minhas experiências com testes de penetração em dispositivos IoT, dispositivos usando vários frameworks eram frequentemente vulneráveis a problemas de segurança básicos. As discussões que tive posteriormente com as equipes de produto revelaram que a mentalidade geral é que, se alguém está usando um framework popular, ele geralmente é considerado seguro por design, resultando em descuido na avaliação de sua segurança.

Não importa de qual lado você esteja, os construtores ou os invasores, é importante observar os problemas de segurança do produto, independentemente da estrutura subjacente ou dos conjuntos de protocolos usados. Por exemplo, você frequentemente encontrará desenvolvedores usando ZigBee pensando que ele é extremamente seguro, deixando seus produtos vulneráveis a todos os tipos de ataques baseados em rádio.

Neste livro, não nos concentramos necessariamente em nenhuma framework ou pilha de tecnologia específica, mas sim em uma abordagem aplicável a qualquer solução de dispositivo IoT, independentemente da arquitetura subjacente. Nesse processo, no entanto, também abordamos alguns protocolos populares (por exemplo, ZigBee e BLE) para lhe dar uma ideia de que tipo de vulnerabilidades esperar e como encontrar esses problemas de segurança.

Algumas das estruturas de IoT mais populares incluem o seguinte:

Eclipse Kura (https://www.eclipse.org/kura/)

The Physical Web (https://google.github.io/physicalweb/)

IBM Bluemix (now IBM Cloud: https://www.ibm.com/cloud/)

Lelylan (http://www.lelylan.com/)

Thing Speak (https://thingspeak.com/)

Bug Labs (https://buglabs.net/)

The thing system (http://thethingsystem.com/)

Open Remote (http://www.openremote.com/)

OpenHAB (https://www.openhab.org/)

Eclipse IoT (https://iot.eclipse.org/)

Node-Red (https://nodered.org/)

Flogo (https://www.flogo.io/)

Kaa IoT (https://www.kaaproject.org/)

Macchina.io (https://macchina.io/)

Zetta (http://www.zettajs.org/)

GE Predix (https://www.ge.com/digital/predixplatform-foundation-digital-industrialapplications)

DeviceHive (https://devicehive.com/)

Distributed Services Architecture (http://iot-dsa.org/)

Open Connectivity Foundation (https://openconnectivity.org/)

Isso é apenas uma pequena fração de alguns dos frameworks de dispositivos IoT mais populares que você encontrará ao mergulhar no mundo da IoT.

Da mesma forma, quando se trata de protocolos de comunicação, há toda uma gama de protocolos sendo usados pelos fabricantes para suas soluções de IoT.

Alguns dos protocolos de comunicação mais populares incluem o seguinte:

- Wi-Fi
- BLE
- Cellular/Long Term Evaluation (LTE)
- ZigBee
- ZWave
- 6LoWPAN
- LoRA
- CoAP
- SigFox
- Neul
- MQTT
- AMQP
- Thread
- LoRaWAN

Para avaliar adequadamente a segurança IoT de um determinado dispositivo ou protocolo de comunicação, você precisará de várias ferramentas de hardware. Por exemplo, o Ubertooth One seria necessário para capturar e analisar pacotes BLE, o Atmel RzRaven para ZigBee, e assim por diante.

Agora que temos uma boa ideia do que é IoT e das várias tecnologias envolvidas, vamos dar uma olhada em alguns dos fatores que levam à insegurança desses dispositivos.

Razões para Vulnerabilidades de Segurança IoT

Dado que os dispositivos IoT são extremamente complexos por natureza, é altamente provável que a maioria dos dispositivos que você encontrar terá problemas de segurança. Se tentarmos entender por que essas vulnerabilidades existem em primeiro lugar, e como você pode evitar esses problemas de segurança ao construir um produto, precisamos nos aprofundar em todo o ciclo de vida de desenvolvimento do produto, desde a fase de ideação até o produto ser lançado no mercado.

Alguns dos motivos que se destacam como causa de problemas de segurança ao construir esses dispositivos são apresentados a seguir.

Falta de Conscientização de Segurança entre Desenvolvedores

Desenvolvedores que trabalham nestes dispositivos inteligentes frequentemente possuem menos conhecimento, ou sequer conhecimento, sobre as possíveis vulnerabilidades de segurança em dispositivos IoT. Considerando que, em grandes organizações, os desenvolvedores geralmente já estão sobrecarregados, seria uma ótima ideia ter reuniões periódicas para discutir como eles podem construir produtos seguros desde o início, incluindo táticas práticas como diretrizes rígidas de codificação a serem seguidas e uma lista de verificação de segurança para qualquer amostra de código em que trabalhem.

Falta de Perspectiva Macro

Como veremos no próximo capítulo sobre os vários componentes que constituem um dispositivo de IoT, é extremamente fácil para desenvolvedores ou equipes de segurança esquecerem o fato de que é a interconexão de dispositivos e diversas tecnologias que pode levar a problemas de segurança. Por exemplo, apenas olhar para o aplicativo móvel pode não revelar problemas de segurança, mas se você combinar as descobertas do aplicativo móvel e como a comunicação de rede funciona, poderá descobrir uma falha crítica de segurança. É essencial que as equipes de produto invistam mais tempo e esforço analisando toda a arquitetura do dispositivo e realizando modelagem de ameaças.

Problemas de segurança baseados na cadeia de suprimentos

Uma das causas das vulnerabilidades de segurança em dispositivos IoT é o envolvimento de muitos participantes. Isso significa que você frequentemente encontrará diferentes componentes de dispositivos sendo fabricados por fornecedores diferentes, tudo sendo montado por outro fornecedor e, finalmente, sendo distribuído por outro ainda. Isso, embora inevitável na maioria das situações, pode levar a problemas de segurança (ou backdoor) que podem ser introduzidos por um deles, colocando todo o produto em risco.

Utilização de Frameworks e Bibliotecas de Terceiros Inseguras

No caso de dispositivos IoT ou qualquer outra tecnologia, é comum encontrar desenvolvedores usando bibliotecas e pacotes existentes, introduzindo amostras de código potencialmente vulneráveis em um produto seguro. Embora algumas organizações tenham verificações de qualidade no código escrito pelos desenvolvedores, muitas vezes elas tendem a negligenciar os pacotes que estão sendo utilizados. Isso também é acompanhado pelas exigências de negócios de uma organização, onde a gerência exige que os produtos cheguem ao mercado em prazos acelerados (geralmente irreais), o que coloca a avaliação de segurança do produto em segundo plano. Muitas vezes, sua importância não é percebida até que o produto sofra uma violação de segurança.

Conclusão

Neste capítulo, vimos o que são dispositivos IoT, os protocolos e frameworks usados por esses dispositivos inteligentes e os motivos pelos quais esses dispositivos geralmente são vulneráveis. Também examinamos alguns dos problemas de segurança identificados anteriormente em soluções populares de dispositivos IoT para entender quais são algumas das vulnerabilidades encontradas em dispositivos do mundo real. No próximo capítulo, examinaremos mais profundamente o mapeamento da superfície de ataque desses dispositivos e como podemos identificar e possivelmente evitar riscos de segurança em dispositivos IoT.

2.Realizando um teste IoT de penetração

Neste capítulo, aprenderemos como realizar um teste de penetração de IoT e entender o primeiro elemento dele, que é o mapeamento da superfície de ataque. Muitos tester's(quem realiza os testes de penetração) ainda não conseguiram migrar para o teste de penetração de IoT devido à falta de conhecimento sobre como realizar um teste de penetração de IoT: Quais são os diferentes componentes envolvidos? Quais ferramentas devem ser usadas? Como você executa um teste geral?

Este capítulo compartilha insights sobre como realizar um teste de penetração de IoT e responder a essas perguntas. Também abordamos a primeira fase do processo de teste de penetração, mapeamento da superfície de ataque, que usamos para avaliar a solução do dispositivo IoT de destino e obter uma estimativa justa de quais tipos de problemas de segurança podem estar presentes no produto que estamos testando.

O que é um teste de penetração de IoT?

Um teste de penetração de IoT é a avaliação e exploração de vários componentes presentes em uma solução de dispositivo IoT para ajudar a tornar o dispositivo mais seguro. Ao contrário dos testes de penetração tradicionais, a IoT envolve vários componentes diferentes, como discutimos anteriormente, e sempre que falamos sobre um teste de penetração de IoT, todos esses componentes precisam ser testados.

Assim como em qualquer teste de penetração de IoT, nós, como profissionais da área, precisamos entender o escopo do teste, bem como quaisquer outras restrições e limitações. As condições do teste de penetração variam de produto para produto e podem incluir qualquer coisa, desde a garantia de que o teste ocorra entre 22h e 5h (ou durante a noite) até a realização do teste em um ambiente de "mentira" fornecido pelo cliente.

Uma vez que você compreenda o escopo técnico do projeto, vale a pena mencionar ao cliente qual tipo de teste de penetração (caixa branca, caixa preta ou caixa cinza) você ou sua equipe irá realizar para garantir que ambos estejam na mesma página.

Um dos outros pontos sobre o teste de penetração de IoT é a necessidade de vários dispositivos. Muitas vezes, durante um teste de penetração de IoT, certas técnicas que usamos envolvem métodos destrutivos, como a remoção de um chip de uma placa de

circuito para análise, o que provavelmente tornaria o dispositivo inutilizável para análises posteriores.

Após as discussões, a próxima etapa é realizar o teste de penetração de acordo com o escopo e a metodologia desejados. Esta fase do teste de penetração começa com o mapeamento de toda a superfície de ataque da solução, seguido pela identificação de vulnerabilidades e realização de exploração, que é então seguida pela pós-exploração. O teste conclui com um relatório técnico aprofundado. Neste capítulo, abordaremos apenas a primeira etapa, o mapeamento da superfície de ataque. Nos próximos capítulos, veremos as várias maneiras de identificar e explorar vulnerabilidades e, no capítulo final, veremos como escrever um relatório de teste de penetração para dispositivos IoT.

Mapeamento de superfície de ataque

O processo de mapeamento da superfície de ataque significa mapear todos os diversos pontos de entrada que um invasor poderia potencialmente explorar em uma solução de dispositivo IoT. Este é o primeiro passo, e um dos mais importantes, em toda a metodologia de teste de penetração de IoT. Ele também envolve a criação de um diagrama de arquitetura de todo o produto da perspectiva do profissional de teste de penetração.

Durante os trabalhos de teste de penetração, frequentemente dedicamos um dia inteiro a esta fase. Esta etapa é útil porque ajuda a entender a arquitetura de toda a solução e, ao mesmo tempo, ajuda a estabelecer vários testes que você executaria no produto, classificados por prioridade. A prioridade dos ataques pode ser determinada pela facilidade de exploração multiplicada pelo impacto da exploração.

Em um caso em que a exploração é extremamente fácil e leva a um comprometimento bem-sucedido e recuperação de dados confidenciais do dispositivo, isso seria classificado como uma vulnerabilidade de alta prioridade e alta criticidade. Por outro lado, algo que é difícil de executar - com resultados obtidos durante o teste que não sejam tão úteis - seria categorizado como uma vulnerabilidade de baixa criticidade e baixa prioridade. Nos trabalhos de teste de penetração, sempre que identificamos uma vulnerabilidade de alta criticidade, também notificamos o fornecedor imediatamente sobre a visão geral da vulnerabilidade e seu impacto no mesmo dia, em vez de esperar a conclusão do trabalho.

Agora que você tem uma noção básica do que fazer no mapeamento da superfície de ataque, vamos nos aprofundar e entender os detalhes exatos de como realizar esse processo.

Como realizar o mapeamento da superfície de ataque

É crucial dedicar tempo para entender completamente o dispositivo antes de iniciar um teste de penetração. Realizar uma avaliação com informações incompletas ou parciais é um dos maiores erros que um profissional de teste de penetração pode cometer. Isso significa vasculhar todos os canais possíveis e coletar informações, como documentação e manuais do dispositivo, recursos online e publicações sobre o produto, e qualquer conteúdo disponível ou pesquisas anteriores sobre o dispositivo.

Anote os vários componentes usados no dispositivo, tipo de arquitetura da CPU, protocolos de comunicação usados, detalhes do aplicativo móvel, processo de atualização de firmware, portas de hardware, suporte a mídia externa nos dispositivos e praticamente qualquer outra coisa que você possa encontrar. Muitas vezes, as coisas não são tão óbvias quanto parecem inicialmente, e é por isso que você deve se aprofundar em cada uma das várias funções que o dispositivo oferece.

Ao analisar uma solução de IoT para mapeamento da superfície de ataque, podemos dividir toda a arquitetura em três categorias:

1. Dispositivo embarcado.
2. Firmware, software e aplicativos.
3. Comunicações de rádio.

Nosso objetivo ao analisar o dispositivo IoT para mapeamento da superfície de ataque é categorizar a funcionalidade e as ameaças de segurança correspondentes a cada categoria. Vamos considerar qual deve ser o processo de pensamento ao categorizar as vulnerabilidades potenciais de acordo com as categorias mencionadas. Cada uma das categorias mencionadas a seguir serve como uma introdução a esse componente e será detalhada com mais profundidade nos próximos capítulos.

Dispositivos incorporados

Um dispositivo embarcado é a peça central de qualquer arquitetura de dispositivo IoT e também é o "objeto" na Internet das Coisas. O dispositivo embarcado em um produto IoT pode ser usado para diversas finalidades, dependendo do cenário de uso. Ele pode funcionar como um hub para toda a arquitetura IoT do dispositivo, como sensor que coleta dados do ambiente físico ao seu redor, ou como forma de exibir dados ou realizar a ação desejada pelo usuário. Assim, os objetos na Internet das Coisas podem ser usados para coletar, monitorar, analisar dados e realizar ações.

Para esclarecer isso com um exemplo do mundo real, pense em um produto IoT para casas inteligentes. Existem muitos dispositivos que juntos formam o produto IoT para casa inteligente. Isso inclui uma porta de entra inteligente ou ponto central, lâmpadas inteligentes, sensores de movimento, interruptores inteligentes e dispositivos conectados adicionais.

Mesmo que os dispositivos tenham finalidades diferentes, na maioria das vezes, a abordagem para testar a segurança desses dispositivos contra vulnerabilidades seria a mesma. Dependendo da finalidade do dispositivo, ele conterá informações confidenciais que, se comprometidas, seriam consideradas críticas.

A seguir estão algumas das vulnerabilidades encontradas em dispositivos embarcados:

- Portas seriais expostas.
- Mecanismo de autenticação inseguro usado nas portas seriais.
- Capacidade de despejar o firmware via JTAG ou chips Flash.
- Ataques baseados em mídia externa.
- Análise de energia e ataques baseados em canais laterais.

Para avaliar a segurança do dispositivo, o processo de reflexão deve ser baseado nessas perguntas: Quais são as funcionalidades do dispositivo? A quais informações o dispositivo tem acesso? Com base nesses dois fatores, podemos estimar realisticamente os potenciais problemas de segurança e seu impacto.

Uma vez que nos aprofundemos na exploração de hardware, no Capítulo 3, entenderemos melhor as falhas subjacentes em dispositivos IoT comuns e veremos como podemos explorar as várias vulnerabilidades de segurança de hardware que encontramos em dispositivos IoT.

Firmware, Software e Aplicativos

Após a exploração do hardware, o próximo componente que analisamos é a parte de software de um dispositivo IoT. Isso inclui tudo, desde o firmware que roda no dispositivo, os aplicativos móveis usados para controlá-lo, os componentes de nuvem conectados a ele e assim por diante.

São também nesses componentes que você pode aplicar a experiência tradicional de pentest ao ecossistema de IoT. Isso também envolveria tópicos como engenharia reversa de binários de diferentes arquiteturas, incluindo Advanced RISC Machines (ARM) e MIPS, bem como engenharia reversa de aplicativos móveis. Esses componentes podem frequentemente ajudá-lo a descobrir muitos segredos e encontrar vulnerabilidades. Dependendo do componente que você está testando, utilizará diferentes conjuntos de ferramentas e técnicas variadas.

Um dos outros objetivos durante o pentest de componentes baseados em software é analisar as várias maneiras pelas quais podemos acessar o componente individual que queremos testar. Por exemplo, se quisermos analisar o firmware em busca de vulnerabilidades, precisaríamos ter acesso ao firmware, o que geralmente não é fácil.

Também precisamos concentrar muitos esforços na engenharia reversa das APIs de comunicação que nos ajudam a entender como os diferentes componentes do dispositivo IoT interagem entre si e a verificar quais tipos de protocolos de comunicação estão em uso.

Se olharmos para um dispositivo IoT do mundo real, uma casa inteligente terá os seguintes componentes que serão abordados na seção de software:

Aplicativo móvel: permite controlar dispositivos inteligentes - ligando e desligando as luzes, adicionando novos dispositivos ao sistema de casa inteligente e assim por diante. Normalmente, você terá aplicativos móveis para as plataformas Android e iOS, que são as duas plataformas de aplicativos móveis dominantes atualmente, esse texto. Existem vários ataques possíveis em aplicativos móveis que podem revelar informações confidenciais do dispositivo ou como ele funciona. Eles também podem servir como um ponto de entrada para atacar o componente web (mencionado posteriormente) por meio da engenharia reversa do binário do aplicativo e de suas APIs de comunicação. Em relação aos aplicativos móveis, também podemos precisar trabalhar com componentes nativos do aplicativo, o que pode nos levar a uma compreensão adicional de todo o binário do aplicativo e várias funcionalidades subjacentes, como criptografia e outros aspectos sensíveis.

Painel baseado na web: Isso permite ao usuário monitorar o dispositivo, visualizar análises e informações de uso, controlar permissões para os dispositivos e assim por diante. A maioria dos dispositivos IoT que você encontrará terá uma interface web onde você pode acessar os dados enviados do dispositivo para o terminal web. Se o aplicativo web for vulnerável, ele pode permitir que você acesse dados não autorizados, que podem ser os dados do mesmo usuário ou de qualquer outro usuário usando o mesmo dispositivo IoT, o que já aconteceu com muitos dispositivos IoT no passado, principalmente monitores de bebês.

Interfaces de rede inseguras são componentes de dispositivos IoT expostos à rede e que podem ser comprometidos devido a vulnerabilidades. Isso pode acontecer de duas formas: por meio de uma porta aberta que aceita conexões a serviços sem qualquer tipo de autenticação, ou por um serviço que esteja executando uma versão vulnerável e desatualizada, com falhas de segurança conhecidas para aquela versão específica.

Já realizamos testes de penetração em diversos dispositivos que executavam versões vulneráveis de componentes como o Simple Network Management Protocol (SNMP - Protocolo Simples de Gerenciamento de Rede) e o File Transfer Protocol (FTP - Protocolo de Transferência de Arquivos).

Firmware: O Coração do Dispositivo

O firmware controla os vários componentes do dispositivo e é responsável por todas as suas ações. Pense nele como o componente que detém as chaves do reino. Praticamente tudo o que você possa imaginar ser extraído do dispositivo pode ser encontrado no firmware. O capítulo dedicado ao firmware neste livro explica o que é firmware, seu funcionamento interno, as várias vulnerabilidades que podemos encontrar e como realizar análises adicionais.

Comunicação entre Dispositivos IoT

Aplicativos móveis, aplicativos web e dispositivos embarcados costumam se comunicar com outros componentes e terminais back-end por meio de diferentes mecanismos de comunicação, como o Representational State Transfer (REST - Transferência de Estado Representacional). Transferência (REST), Simple Object Access Protocol (SOAP-

Protocolo de Acesso Simples a Objetos), Message Queuing Telemetry Transport (MQTT- Transporte de telemetria de enfileiramento de mensagens), Constrained Application Protocol (CoAP- Protocolo de aplicação restrito) e muito mais, que abordaremos brevemente nos próximos capítulos.

Além disso, alguns componentes coletariam dados e os enviariam para um ponto final remoto com frequência, o que muitas vezes poderia ser tratado como uma violação de privacidade, mais apropriadamente, do que um problema de segurança. Todo o foco do mapeamento da superfície de ataque é garantir que você tenha informações suficientes para entender todos os aspectos e funcionalidades do dispositivo, o que nos ajudará a compreender os problemas de segurança neles.

Esses componentes envolvem muitas vulnerabilidades, algumas das quais estão listadas aqui.

Firmware

- Firmware
- Capacidade de modificar o firmware
- Verificação insegura de assinatura e integridade
- Valores confidenciais codificados no firmware - chaves de API, senhas, URLs de “mentira” e assim por diante
- Certificados privados
- Capacidade de entender toda a funcionalidade do dispositivo através do firmware
- Extração do sistema de arquivos do firmware
- Componentes desatualizados com vulnerabilidades conhecidas

Aplicativos móveis

- Engenharia reversa do aplicativo móvel
- Extração do código-fonte do aplicativo móvel
- Verificações inseguras de autenticação e autorização
- Falhas de lógica e negócio
- Vazamento de dados por canal lateral
- Ataques de manipulação de “runtime”
- Comunicação de rede insegura
- Bibliotecas de terceiros e kits de desenvolvimento de software (SDKs) desatualizados

Aplicativo web

- Injeção do lado do cliente
- Referência insegura de objeto direto
- Autenticação e autorização inseguras
- Vazamento de dados confidenciais
- Falhas na lógica de negócios
- Falsificação de requisição entre sites (CSRF)
- “Scripting” entre sites (XSS)

Essa lista é apenas um exemplo de algumas das vulnerabilidades presentes nesses componentes, o que deve lhe dar uma ideia do tipo de vulnerabilidade que afeta esses componentes.

Comunicações de Rádio

As comunicações de rádio fornecem uma maneira para diferentes dispositivos se comunicarem entre si. Essas mídias e protocolos de comunicação geralmente não são considerados pelas empresas ao pensar em segurança, tornando-se, portanto, um ponto ideal para que os testadores de penetração identifiquem vulnerabilidades em dispositivos IoT.

Alguns dos protocolos comuns de comunicação por rádio usados em dispositivos IoT são celular, Wi-Fi, BLE, ZigBee, Wave, 6LoWPAN, LoRa e outros. Dependendo do protocolo de comunicação que um dispositivo está usando, hardware especializado pode ser necessário para realizar a análise da comunicação por rádio.

Durante o processo de análise inicial, você também deve listar todos os diferentes itens de hardware e software necessários para realizar uma avaliação de segurança dos protocolos de rádio em uso. Embora inicialmente possa parecer uma tarefa onerosa, uma vez que você tenha adquirido as ferramentas necessárias para realizar a avaliação, é apenas uma questão de analisar a comunicação usando essas ferramentas.

Configurar software e ferramentas para teste de penetração de rádio (e outros componentes de teste de penetração de IoT) pode ser uma tarefa árdua. É por isso que construímos uma máquina virtual (VM) personalizada chamada AttifyOS que você pode usar para todos os exercícios e laboratórios de teste de peneteação de IoT abordados neste livro. Você pode baixar o AttifyOS em https://www.attify.com/attifyos.

Ao longo deste livro, abordamos três categorias principais em comunicação por rádio que são mais relevantes do ponto de vista de teste de penetração e avaliação de segurança:

Rádio Definido por Software (SDR).

Exploração de ZigBee.

Exploração de BLE (Bluetooth Low Energy).

Dependendo do componente de rádio com o qual estamos trabalhando, ele terá diferentes conjuntos de vulnerabilidades. No entanto, estes são os tipos mais comuns de vulnerabilidades que encontramos em protocolos e meios de comunicação por rádio:

- Ataques Man-in-the-middle.
- Ataques baseados em replay.
- Verificação insegura de Cyclic Redundancy Check (CRC).
- Ataques baseados em jamming.
- Negação de serviço (DoS).

- Falta de criptografia.
- Capacidade de extrair informações confidenciais de pacotes de rádio.
- Interceptação e modificação de comunicação de rádio ao vivo.

Abordaremos essas categorias de ataque e maneiras de realizá-las nos capítulos posteriores deste livro. Ao criar um mapa da superfície de ataque para comunicação por rádio, o processo deve se concentrar nos seguintes itens:

Quais são as funções dos vários componentes envolvidos?

Qual componente inicia o mecanismo de autenticação e emparelhamento?

Como é a aparência do mecanismo de emparelhamento?

Quantos dispositivos cada componente pode manipular simultaneamente?

Em qual frequência o dispositivo opera?

Quais protocolos estão sendo usados por diferentes componentes? Eles são protocolos personalizados ou proprietários?

Existem dispositivos semelhantes operando na mesma faixa de frequência deste dispositivo?

Esses são apenas alguns dos itens que você deve considerar ao analisar a comunicação de rádio para um determinado dispositivo IoT.

Criando um Mapa da Superfície de Ataque

Agora que estamos familiarizados com todos os diferentes componentes que analisaremos e os tipos de vulnerabilidades que afetam os componentes, estamos em uma boa posição para criar um mapa da superfície de ataque de qualquer dispositivo IoT. A Figura 2-2 mostra o processo para criar um mapa da superfície de ataque.

Os passos a seguir descrevem como criar um mapa da superfície de ataque para qualquer dispositivo IoT:

1. Listar todos os componentes presentes no produto alvo.
2. Preparar um diagrama de arquitetura.
3. Rotular os componentes e os fluxos de comunicação entre eles.
4. Identificar vetores de ataque para cada componente e o canal ou protocolo de comunicação utilizado.
5. Categorizar os vetores de ataque com base na criticidade.

O diagrama de arquitetura inicial também nos auxilia durante todo o processo de compreensão da solução IoT e dos diversos componentes envolvidos. Certifique-se de listar todos os componentes envolvidos, por menores que pareçam, juntamente com todas as especificações técnicas de cada componente, durante o processo de criação do diagrama de arquitetura.

Para algumas informações que podem ser difíceis de obter inicialmente, como a frequência em que o dispositivo opera, é possível encontrar dados disponíveis online. Comece por sites como o fccid.io, onde você pode inserir o ID FCC de um dispositivo IoT e encontrar diversas informações sobre ele.

Por exemplo, vamos pegar o kit “Samsung Smart Things”, que consiste em vários dispositivos para automação residencial inteligente. Através de uma análise inicial do site, podemos descobrir que ele contém os seguintes itens:

- Central de Casa Inteligente
- Sensor de movimento
- Tomada inteligente
- Sensor de presença
- Sensor de movimento

Além disso, também possui um aplicativo móvel disponível na Google Play Store e Apple AppStore. O próximo passo é desenhar um diagrama desses componentes para nos ajudar a visualizá-los melhor. A Figura 2-3 é um exemplo de diagrama de arquitetura que criei para um dispositivo doméstico inteligente de amostra.

Os seguintes componentes estão envolvidos neste sistema de casa inteligente:

- Dispositivos.
- Aplicativo móvel.
- Gateway IoT.
- Recursos de nuvem.
- Protocolos de comunicação: BLE, Wi-Fi, ZigBee, ZWave, 6LoWPAN, GSM e Ethernet.
- Os dispositivos e o aplicativo móvel se comunicam via BLE.
- O hub inteligente e os dispositivos se comunicam por meio de vários protocolos, como ZigBee, ZWave e 6LoWPAN.
- O aplicativo móvel também pode interagir com o hub inteligente via Wi-Fi.
- O aplicativo móvel e o hub inteligente se comunicam com a nuvem a cada cinco minutos e compartilham dados.
- O aplicativo móvel usa uma API REST para se comunicar por meio dos recursos de nuvem.
- Podemos ver esses detalhes adicionais especificados na Figura 2-3:
    - O gateway do hub inteligente possui uma porta Ethernet e um slot para cartão SD externo que pode ser usado para atualização de firmware.
    - O dispositivo contém um processador Broadcom.
    - O aplicativo móvel é um aplicativo nativo com a possibilidade de possuir bibliotecas adicionais.
- Durante o processo de configuração inicial, o dispositivo é configurado com uma senha padrão de "admin".
- O aplicativo Android ainda funciona se houver um problema de certificado; ou seja, o aplicativo funciona em conexões inseguras com autoridades certificadoras (CAs) não confiáveis para o certificado SSL.

Como você pode ver na Figura 2-3, temos todos os diferentes componentes mencionados no diagrama, juntamente com os vários canais de comunicação e protocolos que os vários dispositivos usam para se comunicar entre si ou com os terminais da web.

Depois de examinar este diagrama e todas as especificações técnicas, quando iniciamos nosso teste de penetração, agora sabemos exatamente como abordar esses dispositivos e quais são nossos alvos detalhados. Nesta etapa, precisamos pensar como um atacante. Se você tivesse que atacar um componente, como faria isso?

Quais vulnerabilidades você procuraria? Quais casos de teste você realizaria? Explorar esse componente específico é em que você deve se concentrar.

Com base em todas essas informações, prepare uma planilha com todos os casos de teste e explorações a serem testados nos diversos componentes, incluindo uma descrição detalhada de qual teste específico você realizará e qual será a saída se o ataque for bem-sucedido. Quanto mais detalhada for a sua planilha, mais eficaz será o seu teste de penetração. Se você estiver trabalhando em equipe, esta planilha é algo que você deve debater com sua equipe e depois ajustar. A Figura 2-4 mostra uma planilha de amostra.

Você também pode aproveitar os recursos disponíveis em vários lugares, incluindo:

- Guia de teste de penetração de IoT da Attify disponível em http://www.iotpentestingguide.com (esse link pode ser removido se você não quiser incluir o endereço).
- Guia de Hacking Embarcado pela OWASP.
- Superfície de Ataque IoT da OWASP.

Estruturação do teste de penetração

Como o teste de penetração de IoT é relativamente novo em comparação com outras formas de teste de penetração, poucas pessoas estão familiarizadas com a forma de executar o teste de penetração completo. Esta seção explica como estruturar o teste de penetração, o tamanho ideal da equipe, o número de dias necessários e outros detalhes relevantes.

Novamente, tudo isso vem da experiência pessoal de testes de penetração em centenas de dispositivos IoT nos últimos anos - e de encontrar problemas críticos de segurança em quase todos eles. Acredito que essa abordagem funcione de forma eficiente. Se você tiver outra abordagem para executar o teste de penetração que funcione melhor para você, certamente poderá continuar com ela.

Em seguida, a estrutura geral de um teste de penetração de IoT é explicada em detalhes.

Engajamento do Cliente e Ligação Inicial para Discussão

Esta é a ligação inicial para discussão após recebermos uma solicitação de uma organização para realizar um teste de penetração em seu dispositivo IoT. Mesmo antes desta etapa, temos uma discussão inicial com nossa equipe técnica para verificar se temos experiência relevante para o dispositivo IoT em questão e outros requisitos logísticos - recursos disponíveis, próximas datas disponíveis e assim por diante.

Durante esta etapa, colocamos nosso líder de teste de penetração em uma ligação com o cliente para discutir o dispositivo. Aqui estão algumas das perguntas que abordamos: Qual é o resultado esperado do teste de penetração? Quais componentes eles querem focar mais? Eles gostariam de um teste normal ou um teste de penetração com uma equipe de pesquisa adicional envolvida?

Se você é um tester’s(profissional que realiza testes), não posso enfatizar demais que seus clientes são seus ativos mais valiosos; é extremamente importante que você forneça serviços e ofertas apenas no domínio em que você e sua equipe se destacam. Dessa forma, você poderá atender melhor o cliente e construir um relacionamento duradouro.

Discussão técnica adicional e chamada informativa

Uma vez que decidimos que este é o projeto em que queremos trabalhar, e seríamos capazes de agregar valor ao engajamento geral com uma ótima pesquisa, pedimos ao cliente que sua equipe técnica participe de uma discussão com nossa equipe de teste de penetração que trabalhará nesse engajamento. Lembre-se, esta etapa ocorre após a assinatura de um acordo de confidencialidade e outra documentação necessária para que o cliente possa compartilhar livremente as especificações técnicas do produto.

Fazemos muitas perguntas durante esta fase para entender melhor o produto. Isso nos permite entender melhor o produto e explicar ao cliente nossa metodologia de teste de penetração e o que eles podem esperar durante cada etapa do teste de penetração. Também compartilhamos nosso mecanismo de relatório seguro, sistema de relatório diário, casos de teste que realizaremos, equipe de avaliação, interagindo com seu mecanismo de back-end e assim por diante. É importante ser transparente e justo com o cliente em relação à sua metodologia de teste de penetração e aos resultados que eles devem esperar a cada dia, ao final de cada fase e ao final do engajamento.

Também precisamos entender o processo de desenvolvimento deles, que tipo de teste sua equipe de segurança executa, se o teste de garantia de qualidade (QA) envolve testes de segurança, se existe um ciclo de vida de desenvolvimento seguro e assim por diante. Isso também ajuda a apresentar as equipes umas às outras, pois também oferecemos suporte personalizado aos desenvolvedores quando eles estão corrigindo as vulnerabilidades.

Obviamente, a maioria desses componentes corresponde a uma avaliação de caixa cinza, mas você entende a ideia. Durante um teste de penetração de caixa preta, você omitiria detalhes que um invasor não teria, o que também é chamado de simulação de exploração de atacante. Uma simulação de exploração de atacante é um método de teste

de penetração no qual você compromete e ataca o dispositivo final da maneira que um atacante altamente direcionado faria.

Explotação Simulada de Ataque

Esta é a fase real de teste de penetração onde encontramos vulnerabilidades em produtos de IoT e as exploramos. Uma vez que recebemos os dispositivos em nossos laboratórios, nosso processo de teste de penetração é executado paralelamente com diversas atividades acontecendo ao mesmo tempo: nossa equipe de engenharia reversa trabalha na engenharia reversa de vários binários, a equipe de hacking embarcado invade o dispositivo de hardware IoT, a equipe de Rádio Definido por Software (SDR) trabalha na exploração da comunicação via rádio e a equipe de teste de penetração de software trabalha em firmware, aplicativos móveis, aplicativos web e ativos baseados em nuvem.

Isso só é possível se você tiver uma equipe forte, com diferentes divisões de teste de penetração e indivíduos com expertise em sua área de atuação. Se você é um pesquisador de segurança individual, também pode fazer isso, mas para teste de penetração de IoT, eu recomendo fortemente construir uma equipe de pelo menos três pessoas - especialistas em software e firmware, hardware e rádio - antes de realizar testes de penetração.

Uma vez que o engajamento é concluído, compartilhamos um relatório altamente detalhado junto com scripts de PoC, demonstrações em vídeo de alta qualidade, técnicas usadas para encontrar as vulnerabilidades, etapas para reprodução, métodos de correção e referências adicionais que fornecem mais informações sobre as vulnerabilidades identificadas.

Remediação

Após concluirmos o teste de penetração, trabalhamos com os desenvolvedores, oferecendo suporte por voz, videochamada e e-mail, identificando exatamente o que precisa ser mudado e quais patches precisam ser implementados. Embora todas essas informações sejam fornecidas no relatório técnico, descobrimos que trabalhar com os desenvolvedores durante esta fase e oferecer suporte os ajuda a corrigir bugs com mais rapidez e a evitar cometer os mesmos erros novamente, graças ao aprendizado com nossa equipe durante as discussões de remediação.

Revalidação

Uma vez que as vulnerabilidades de segurança tenham sido corrigidas pelos desenvolvedores, realizamos outro teste de penetração para as vulnerabilidades identificadas no teste inicial. Isso garante que todos os patches estejam instalados e que os patches aplicados pelos desenvolvedores sejam seguros e não causem vulnerabilidades em outros componentes. Esse é um dos erros que vemos os testadores de penetração cometerem: uma vez que o dispositivo é corrigido, eles limitam o teste de revalidação apenas aos componentes que consideraram vulneráveis. No entanto, é preciso prestar muita atenção para garantir que a correção do código naquele local não tenha levado à criação de bugs em outro. É assim que concluímos nosso teste de penetração para aquela versão do dispositivo.

Conclusão

Neste capítulo, aprendemos como iniciar um teste de penetração de IoT, criando um modelo de ameaça, também conhecido como mapeamento da superfície de ataque para o produto. Também exploramos o que há por trás da superfície e demos uma olhada nos vários componentes presentes em uma arquitetura de IoT e as vulnerabilidades de segurança que poderíamos encontrar nesses componentes.

Ponto de ação

1. Pegue qualquer dispositivo IoT ao seu redor (ou pense em um) e crie um diagrama da arquitetura desse dispositivo.
2. Depois de criar o diagrama da arquitetura, adicione os detalhes de como os dispositivos interagem entre si: quais componentes se conectam a quais e qual meio de comunicação e protocolo estão sendo usados.
3. Liste os problemas de segurança correspondentes a cada nó e a cada meio no diagrama que você criou.

Para receber feedback sobre sua criação e processo de pensamento, envie-me um e-mail com uma foto do seu diagrama e quaisquer anotações adicionais para:

iothandbook@attify.com

3.Anasilando Hardware

Este é provavelmente o capítulo mais importante para você se nunca mexeu com hardware antes. Neste capítulo, veremos como podemos entender o hardware de um dispositivo IoT do ponto de vista da segurança, tanto para análise interna quanto externa. O dispositivo, como vimos nos capítulos anteriores, é um dos principais

componentes em qualquer produto IoT. É o componente do dispositivo que pode ajudar a revelar muitos segredos sobre o dispositivo para nós, o que também veremos mais adiante neste capítulo.

A realização de análises de hardware pode ajudá-lo com as seguintes tarefas:

- Extrair firmware do dispositivo IoT do mundo real.
- Ganhar “root shell” no dispositivo para obter acesso irrestrito.
- Executar depuração ao vivo para ignorar proteções e restrições de segurança.
- Gravar novo firmware no dispositivo.
- Estender a funcionalidade do dispositivo.

Em alguns casos, abrir um dispositivo pode fazer com que ele não funcione corretamente (devido a violação física) ou você não conseguir remontá-lo. É por isso que, sempre que você estiver realizando um teste de penetração de dispositivo IoT, deve sempre pedir dois (ou mais) conjuntos de dispositivos ao cliente para que possa realizar avaliações de segurança física em um deles e o restante dos testes de vulnerabilidade no outro.

Se você nunca abriu hardware antes, tome cuidado especial ao trabalhar com os procedimentos deste capítulo para não se machucar. Seja sempre gentil e descubra uma maneira de abrir o dispositivo com cuidado para que possa continuar usando-o posteriormente, você pode colocá-lo de volta inteiro depois da análise. Agora, vamos começar.

Inspeção Externa

O primeiro passo na análise física do dispositivo é realizar uma inspeção externa. Isso inclui fazer um exame básico do dispositivo observando seus vários aspectos, incluindo itens como:

- Quantos e quais botões estão presentes.
- Opções de interface externa: porta Ethernet, slot para cartão SD e muito mais.
- Qual tipo de display o dispositivo possui.
- Requisitos de energia e voltagem para o dispositivo.
- Se o dispositivo possui alguma certificação e o que elas significam.
- Quais etiquetas de identificação FCC estão na parte traseira.
- Que tipo de parafusos o dispositivo usa.
- Se o dispositivo se parece com outros dispositivos com funcionalidades semelhantes que você já viu no mercado (talvez seja apenas um modelo renomeado).
- E assim por diante (você entende a ideia!).

Esta análise inicial lhe dará uma melhor compreensão do dispositivo como um todo e de como ele funciona, ao mesmo tempo que o ajudará a entender alguns detalhes internos do dispositivo.

Antes mesmo de abrir o dispositivo, há algumas coisas que você pode fazer apenas realizando esta análise inicial. A análise inicial normalmente envolve uma inspeção visual do dispositivo e uma revisão de outras fontes de informação sobre ele. Esta etapa também envolve usar o dispositivo e descobrir qual é sua funcionalidade normal. Depois de determinar a funcionalidade normal do dispositivo, você poderá definir abordagens direcionadas para subverter sua funcionalidade.

Trabalhando com um Dispositivo Real

Vamos pegar um dispositivo de amostra e começar a observá-lo conforme descrito. Neste caso, o dispositivo é um sistema de navegação e o modelo é Navman N40i. Apenas com uma rápida pesquisa inicial no Google, você pode aprender várias especificações do dispositivo, como esses:

1. Ele roda o Windows CE 5.0.
2. Possui uma câmera de 1.3 MP.
3. Oferece cinco horas de bateria.
4. Possui um processador Samsung 2400 de 400 MHz.
5. Possui 64 MB de SDRAM.
6. Possui 256 MB de ROM.
7. Contém um chip GPS SiRF STAR II. Estas informações úteis serão valiosas se mais tarde decidirmos encontrar vulnerabilidades no sistema Navman. Este rápido exemplo ilustra como abordar seu dispositivo assim que o obtiver e a análise inicial que você deve realizar.

Identificando Portas de Entrada e Saída O próximo passo é entender como funciona a entrada e saída (I/O) do dispositivo e o número de portas de E/S e outras conexões. Na Figura 3-1, podemos ver que o sistema Navman consiste em uma tela de 3,5 polegadas com cinco botões na frente, juntamente com um LED indicador na esquerda.

É assim que realizamos uma inspeção visual externa de um determinado dispositivo IoT. Lembre-se, esta é apenas a primeira etapa da análise do hardware. Para realizar uma boa inspeção, você precisa analisar os componentes externos e internos, juntamente com a criação de um mapa da superfície de ataque, conforme discutido no Capítulo 2.

Inspeção Interna

Após a inspeção externa, passamos para a inspeção interna. Como o próprio nome sugere, isso envolve abrir o dispositivo e observar seus componentes internos para melhor compreendê-lo e identificar as possíveis superfícies de ataque.

Para abrir o dispositivo, precisamos desapertar os parafusos. Os dispositivos IoT podem ter vários tipos de parafusos e, frequentemente, um conjunto de chaves de fenda comum não consegue abrir alguns dos tipos menos comuns encontrados nos dispositivos. Certifique-se de ter um bom conjunto de chaves de fenda à mão sempre que realizar exploração de IoT no dispositivo. Além disso, tome cuidado especial ao abrir o

dispositivo para não danificar e seus circuitos internos. Um dos erros mais comuns que vejo as pessoas cometerem é tentar forçar a abertura do dispositivo, o que geralmente leva a danos físicos ao dispositivo, em alguns casos tornando-o não funcional.

Se você estiver vendo o interior do dispositivo pela primeira vez na vida, pode ficar fascinado com o que verá. O interior do dispositivo geralmente envolve muitos componentes, incluindo a placa de circuito impresso (PCB), conectores, antena, periféricos e assim por diante. Tente abrir o dispositivo alvo com cuidado e remover todos os cabos, fitas ou quaisquer outros periféricos conectados, um por um (veja a Figura 3-6).

Se você observar atentamente a Figura 3-6, poderá ver um módulo de câmera, uma bateria, um fone de ouvido, um conector GPS e uma antena GPS na parte superior com um LCD conectado por um cabo flexível. Se você olhar para o outro lado da placa, a Figura 3-7 mostra o que você verá.

Agora vamos analisar os diferentes componentes e suas funcionalidades, começando pelo processador. A Figura 3-8 mostra o processador usado em nosso sistema de navegação.

O processador é um dos componentes vitais de qualquer dispositivo IoT. Neste caso, o processador utilizado é o S3C2440AL, que é um processador Samsung ARM. Se pesquisarmos online por S3C2440AL, poderemos encontrar a folha de dados do processador, o que nos levaria a informações mais detalhadas sobre ele (Figura 3-9). A folha de dados deste processador pode ser encontrada em https://www.keil.com/dd/docs/datashts/samsung/s3c2440_um.pdf. Ele contém informações como portas de E/S, interrupções, Relógio de Tempo Real (RTC), Interface Serial Periférica (SPI) e muito mais.

Em seguida, podemos examinar outros componentes, como SDRAM e ROM, que estão presentes no dispositivo, conforme mostrado na Figura 3-10.

Na Figura 3-10, os componentes usados possuem os números K4M561633G, o que, pesquisando online, podemos ver que é uma SDRAM Móvel de 4M × 16 bits × 4 Bancos da Future Electronics, e também podemos ver 512 MB de ROM nele. Continuando, podemos continuar procurando por diferentes componentes - identificando seus números de peça e depois consultando-os online para aprender mais sobre eles.

Uma outra maneira de identificar componentes é observando seus logotipos e consultando um catálogo de referência online, como https://www.westfloridacomponents.com/.

Para procurar folhas de dados, você pode simplesmente pesquisar online pelo número do componente ou visitar um dos sites que contêm catálogos de folhas de dados, como https://www.alldatasheet.com/ ou https://www.datasheets360.com/.

Há um último ponto que ainda não vimos, que provavelmente é o aspecto mais importante: as portas e interfaces de debug. Frequentemente, os dispositivos expõem interfaces de comunicação que podem ser exploradas para obter acesso adicional ao

dispositivo para realizar ações como ler os logs de debug ou até mesmo obter um shell root não autenticado no dispositivo alvo. Como você pode ver na Figura 3-11, em nosso dispositivo, temos o dispositivo expondo interfaces que poderíamos usar para nos comunicar com o dispositivo usando UART e JTAG.

Essas interfaces podem ser encontradas apenas olhando para a PCB e identificando Tx e Rx para UART e TRST, TMS, TDO, TDI e TCK para JTAG, ambos os quais abordaremos detalhadamente nos próximos capítulos. Se você não estiver familiarizado com esses termos, não se preocupe, pois é aqui que passaremos a maior parte do nosso tempo de hacking de hardware no restante do livro.

Analisando Folhas de Dados Muitos dispositivos podem não ter muitas informações técnicas disponíveis em seu site oficial. É aqui que o banco de dados de ID da FCC vem em socorro. Se você é um engenheiro eletrônico e deseja se aprofundar no dispositivo e talvez até olhar para os esquemas do dispositivo, onde você iria? O banco de dados da FCC é a resposta. Então, o que é o banco de dados da FCC, você pode perguntar.

O que é o ID FCC?

A Comissão Federal de Comunicações (FCC) é um órgão regulador responsável por diversos dispositivos que emitem sinais de rádio (a maioria dos dispositivos de IoT se encaixa nessa categoria). A regulamentação existe porque o espectro de rádio é limitado e há aparelhos operando em frequências diferentes.

Sem um órgão regulador, um fabricante poderia escolher qualquer frequência para o seu dispositivo, mesmo que já estivesse sendo usada por outro, causando interferência na comunicação de outros equipamentos.

Portanto, qualquer dispositivo que utilize comunicação via rádio precisa passar por um processo de aprovação, que envolve diversos testes. Após a aprovação, a FCC concede um código de identificação exclusivo para o dispositivo. O ID FCC é o mesmo para um determinado modelo de um fabricante específico. É importante ressaltar que o ID FCC não é uma permissão para transmissão, mas sim a aprovação de uma agência reguladora do governo dos Estados Unidos.

Você pode encontrar o ID FCC do dispositivo impresso nele ou pesquisando online em diferentes fontes. Não confunda o ID FCC com dispositivos que apenas cumprem as regulamentações da FCC, pois eles podem não precisar de um ID FCC, já que não se comunicam sem fio e geram apenas pequenas quantidades de ruído de rádio não intencional.

As informações sobre o processo de teste estão disponíveis no site da FCC, a menos que o fabricante solicite confidencialidade do documento. Você pode pesquisar informações sobre um dispositivo usando o ID FCC no site oficial da FCC em https://www.fcc.gov/oet/ea/fccid ou em sites não-oficiais como fccid.io ou fcc.io.

Usando o ID FCC para encontrar informações sobre o dispositivo

Vamos pegar um dispositivo comercial real e usar o ID FCC para encontrar informações sobre ele. Nesse caso, usaremos a câmera IP Edimax 3116W, controlável por aplicativos móveis e web.

A Figura 3-12 mostra a aparência do dispositivo. Observe o ID da FCC na etiqueta na parte de trás.

Se consultarmos o ID FCC do dispositivo, que é NDD9530401309, no site [fccid.io], veremos a tela mostrada na Figura 3-13.

No site, podemos ver várias informações sobre o dispositivo, como alcance de frequência, acesso à configuração do laboratório, fotos internas, fotos externas, manual do usuário, procuração eletrônica (PoA) e muito mais.

Uma das coisas mais interessantes para se observar ao analisar as informações do FCC ID são as fotos internas do dispositivo. Você pode encontrá-los em https://fccid.io/.

A Figura 3-14 mostra as fotos internas do dispositivo. Um fato interessante a se notar é que, neste caso, as fotos também revelam que esta câmera IP possui uma interface UART, conforme sugerem os quatro “pads” mostrados na foto. Isso também é algo que exploraremos em nossos próximos capítulos para obter um shell root no dispositivo.

Como podemos ver, os IDs da FCC podem ser uma mina de ouro de informações e podem nos ajudar a aprender muitos detalhes sobre o dispositivo e seu funcionamento.

Outro fato interessante é que, às vezes, o fabricante deixa de solicitar um pedido de confidencialidade sobre informações confidenciais do dispositivo, como o esquema elétrico. O acesso ao esquema do dispositivo é extremamente útil, pois nos diz quais componentes eletrônicos diferentes são usados para construí-lo e nos ajuda a entender o dispositivo com muito mais profundidade.

Pacote de Componente

Um dos pontos que vale a pena mencionar, sempre que discutimos análise embarcada ou de hardware, é o tipo de encapsulamento. Sempre que você olhar para o interior de um dispositivo, verá vários componentes diferentes. Cada um dos componentes varia em tamanho, formato e outros aspectos com base na característica e funcionalidade do dispositivo.

Durante a fabricação e desenvolvimento de um dispositivo embarcado, existem várias opções de encapsulamento à sua escolha. Com base no encapsulamento que um componente está usando, para análise, precisaríamos de adaptadores de hardware correspondentes e outros componentes para interagir com eles. Os tipos de

encapsulamento usados com mais frequência estão listados aqui e mostrados na Figura 3-15.

1. DIL

Pacote único em linha

Pacote duplo em linha

TO-220

2. SMD

Cerpack

BGA

SOT-23

QFP

SOIC

SOP

Chipsets de Rádio

Uma das coisas adicionais importantes que você pode procurar nos dispositivos são os vários chipsets de rádio presentes. Esses chipsets podem lhe dar uma ideia de quais tipos de metodologias de comunicação um determinado dispositivo usa, mesmo que não esteja documentado ou mencionado em nenhum lugar.

Por exemplo, a Figura 3-16 é uma imagem interna de um Wink Hub que usa vários protocolos de comunicação, incluindo Wi-Fi, ZigBee e ZWave, além das interfaces de comunicação de hardware, como JTAG. [Imagem do Wink Hub mostrando os chips de rádio] (Figura 3-16 Wink Hub - chips de rádio)

Conclusão

Exploramos componentes de hardware adicionais em detalhes à medida que avançamos neste livro. No entanto, para um conhecimento profundo de vários componentes de hardware, eu recomendo fortemente que você dê uma olhada no livro Hardware Hacking de Nicholas Collins, disponível em http://www.nicolascollins.com/texts/originalhackin gmanual.pdf