UNIVERSIDADE PAULISTA - UNIP

TIEFERSON LEANDRO DOMINGOS

ANÁLISE DE VULNERABILIDADE EM SISTEMAS DE CFTV

LIMEIRA
2017

UNIVERSIDADE PAULISTA - UNIP

TIEFERSON LEANDRO DOMINGOS

ANÁLISE DE VULNERABILIDADE EM SISTEMAS DE CFTV

Trabalho de conclusão de curso apresentado à banca examinadora da Faculdade de Ciências da Computação da UNIP, como requisito à obtenção do grau de Bacharel em Ciências da Computação sob a orientação do professor Mestre Antonio Mateus Locci e professor Mestre Sérgio Nunes.

LIMEIRA
2017

TIEFERSON LEANDRO DOMINGOS

ANÁLISE DE VULNERABILIDADE EM SISTEMAS DE CFTV

Trabalho de conclusão de curso apresentado à banca examinadora da Faculdade de Ciência da Computação da Universidade Paulista (UNIP), como requisito à obtenção do grau de Bacharel em Ciência da Computação sob a orientação do professor Mestre Antonio Mateus Locci e professor Mestre Sérgio Nunes.

Aprovada em __ de ____ de 2017.

BANCA EXAMINADORA

_____

Prof. Dr. Nome completo

_____

Prof. Me. Nome completo

_____

Prof. Me. Nome completo

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus por não me punir por minhas blafêmias, à minha família pela formação moral e ética, aos meus amigos pelo constante compartillhamento de ideias e conhecimento, e aos meus professores pela grande paciência.

“Nunca diga às pessoas como fazer as coisas. Diga-lhes o que deve ser feito e elas o surpreenderão com sua engenhosidade.” George Patton

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

RESUMO

Sistemas de CFTV são desenvolvidos para fornecer segurança e conforto à seus utilizadores, mas o que acontece quando a segurança destes equipamentos é deixada nas mãos dos usuários, ou pessoas sem a devida capacitação? A má configuração e o uso de senhas fracas pode comprometer seriamente sistemas inteiros permitindo que indivíduos denominados crackers acessem informações confidenciais e façam uso destas em benefício próprio. Através de técnicas relativamente simples como o brute force e o port scan, é possível obter acesso à grandes quantidades de sistemas de monitoramento e explorá-los da forma desejada, para o bem ou para o mal. Verifica-se, ao final, uma quantidade considerável de dispositivos de CFTV expostos na Internet e, disponíveis para acesso, comprometendo a privacidade de pessoas e empresas. Observa-se assim a necessidade de investimento na capacitação e conscientização de todos os envolvidos no processo de segurança.

Palavras-Chave: Segurança, CFTV, senhas, pentest, hacker, cracker, brute force

ABSTRACT

CCTV systems are designed to provide security and comfort to your users, but what happens when the security of these devices is left in the hands of the users, or people without the necessary training? Poor configuration and the use of weak passwords can seriously compromise entire systems by allowing individuals called crackers to access sensitive information and make use of it for their own benefit. Through relatively simple techniques like brute force and port scan, it is possible to gain access to large amounts of monitoring systems and exploit them in the desired way, for better or for worse. In the end, we can see a significant number of CCTV devices exposed on the Internet and available for access, compromising the privacy of individuals and companies. The need for investment in the training and awareness of all those involved in the security process is observed.

Keywords: Security, CCTV, passwords, pentest, hacker, cracker, brute force

SUMÁRIO

1. INTRODUÇÃO

Em 2014 o site Russo Insecam[1], numa tentativa de expor a falta de segurança em dispositivos de CFTV[2], começou a compartilhar imagens de câmeras conectadas à internet com qualquer visitante . “O site apresenta *feeds* ao vivo de famílias e empresas em todo o mundo, incluindo um ginásio em Manchester, um quarto em Birmingham e um escritório em Leicester”. (WEAVER, 2014)

"O site, baseado na Rússia, acessa as informações usando as credenciais de login padrão, que estão gratuitamente disponíveis on-line, para milhares de câmeras.” (RICE, 2014 apud WEAVER, 2014). Este trabalho, assim como o *Insecam*, visa alertar sobre o problema do uso de credenciais padrão e senhas inseguras em dispositivos de CFTV, mas sem expor a privacidade de terceiros.

É detalhado o método utilizado para a aquisição da lista de dispositivos que foram testados, bem como a técnica utilizada para a verificação de vulnerabilidade. Por questões legais, o código do *software* utilizado para o teste de acesso será omitido.

Conforme Schneier (2004 p. 17), “As violações da privacidade podem facilmente levar a fraudes” e ainda complementa que criminosos podem usar sistemas de alarme e monitoramento para lhes dar detalhes atualizados sobre a ocupação de estabelecimentos.”Onde quer que os dados possam ser explorados, alguém vai tentar, computadores ou não computadores.” (SCHNEIER, 2004 p. 17)

---

[1] http://www.insecam.org

[2] O CFTV é um sistema de monitoramento de segurança fechado e em vídeos, de alta resolução, onde as imagens das câmeras podem ser visualizadas em um monitore e os arquivos de imagens ficam armazenados em um aparelho de arquivo chamado DVR (RIBEIRO, 2016)

Desta forma, este trabalho objetiva demonstrar quão inseguros e frágeis são os sistemas de monitoramento hoje existentes, quando a segurança dos mesmos é deixada nas mãos do usuário, e propor contramedidas para evitar a exposição da privacidade de empresas e pessoas na Internet.

Schneier (2004, p. 18), afirma que, embora os ataques virtuais tenham grande semelhança com os do mundo físico, a *Internet* tem três novas características que tornam os ataques virtuais piores. 1) Automação: o invasor pode realizar ataques usando programas que executam os passos de forma automática. 2) Ação à distância: um criminoso pode invadir um sistema localizado em Paris a partir de um computador em São Paulo. 3) Propagação da técnica: assim que um invasor descobre uma brecha de segurança ele compartilha detalhes de como explorá-la com outros na *Internet*.

As três características podem ser comprovadas nesta leitura, uma vez que, para a obtenção dos dispositivos e teste de segurança, foi utilizado um programa codificado para este fim (Automação). Estas tarefas foram executas a partir de um computador no estado de São Paulo, contra equipamentos localizados em todo território nacional (Ação à distância) e, através deste trabalho a técnica pode ser reproduzida (Propagação da Técnica).

2. METODOLOGIA

2.1. Hackers x Crackers

Quando se fala em segurança da informação logo imaginamos a figura de uma pessoa de capuz sentada em frente ao computador num quarto escuro e pejorativamente conhecida como Hacker.

> [...] naquela época usávamos o termo hacker para descrever uma pessoa que passava grande parte do tempo mexendo com hardware e software, seja para o desenvolvimento de programas mais eficientes, seja para eliminar etapas desnecessárias e fazer um trabalho mais rapidamente. O termo agora se tornou pejorativo com o significado de "criminoso malicioso". Uso o termo como sempre o usei --- no seu sentido mais antigo e benigno. (MITNICK, 2003)

Hacker e cracker são denominações utilizadas no mundo da tecnologia para separar dois grupos de indivíduos com habilidades semelhantes, porém com intenções distintas. Conforme Ferreira (2014),

> De forma geral, os hackers podem ser vistos como pessoas que têm grande conhecimento sobre sistemas, hardware e redes de computadores utilizando este conhecimento para encontrar falhas e vulnerabilidades com o intuito de melhorar sistemas e impedir que vazamentos de informações ou falhas graves possam acontecer. Diferentemente, os crackers, que possuem os mesmos conhecimentos de um hacker, utilizam-se de técnicas para invadir sistemas, roubar informações e comprometer o funcionamento dos mesmos.

Existem aqueles que discordam dos termos hacker e cracker com base nas categorias existentes para hackers: White Hat (Chapéu branco), Grey Hat (Chapéu cinza) e Black Hat (Chapéu preto).

O chapéu branco é aquele que utiliza seus conhecimentos de forma lícita e ética para elaborar e modificar softwares e hardwares de computadores (Eric Raymond apud FERREIRA, 2014, p. 1). O chapéu cinza tem as mesmas intenções do chapéu branco, mas suas ações são eticamente questionáveis. Já o chapéu preto, foca suas energias em quebrar sistemas, roubar informações e cometer crimes utilizando a tecnologia. Furtos digitais, roubo de senhas e desenvolvimento de softwares maliciosos são alguns exemplos de atividades deste grupo (FERREIRA , 2014, p. 1).

Utilizaremos apenas os nomes Hacker e Cracker para separar estes grupos, mas mesmo com as melhores intenções, perante a lei, Hackers ainda cometem crimes por se utilizarem de técnicas de invasão, muitas vezes, sem o consentimento do proprietário do sistema.

2.2. Lei nº 12.737

A Lei 12.737 acrescenta ao Código Penal, os artigos 154-A e 154-B que tipificam como crime a invasão de dispositivo informático e a interrupção ou perturbação de serviços de tecnologia.

Assim o art. 154-A dispõe:

> Art. 154-A. Invadir dispositivo informático alheio, conectado ou não à rede de computadores, mediante violação indevida de mecanismo de segurança e com o fim de obter, adulterar ou destruir dados ou informações sem autorização expressa ou tácita do titular do dispositivo ou instalar vulnerabilidades para obter vantagem ilícita

E o parágrafo 1º acrescenta:

> §1º Na mesma pena incorre quem produz, oferece, distribui, vende ou difunde dispositivo ou programa de computador com o intuito de permitir a prática da conduta definida no caput.

Desta forma, com base na referida lei, este trabalho se ateve à execução do teste de acesso à dispositivos utilizando credenciais padrão, sem de fato visualizar o conteúdo destes sistemas, assim esta análise é apenas de caráter estatístico mostrando, ao final, números que comprovam a existência da vulnerabilidade que este se propõe a estudar, mas sem comprometer a privacidade alheia. Da mesma forma, não será publicado o código fonte do programa que realiza o teste de vulnerabilidade.

2.3. CFTV

CFTV, ou Circuito Fechado de TV, é um sistema de monitoramento que permite o gerenciamento de uma certa quantidade de câmeras

através de um ponto central conhecido como DVR[3], sendo este também o responsável pelo armazenamento das gravações por um período de tempo determinado. Estes equipamentos são geralmente conectados a um *link* de *Internet* permitindo acesso às imagens remotamente através de aplicativos de celular ou a partir de computadores usando um navegador de *Internet* ou um *software* específico.

Como todo sistema que possui informações confidenciais ou sensíveis, neste caso imagens, seja de pessoas ou bens, é necessária a autenticação do usuário para acesso ao sistema, e esta se dá na forma do conjunto usuário e senha conhecido também como credenciais de acesso. Conforme Abreu (2011, p. 16) “Uma senha é um mecanismo de autenticação, utilizada no processo de verificação de identidade do usuário, assegurando que este é quem realmente diz ser.” Assim, é necessário que o usuário, ou *username*, identifique aquele que está acessando o sistema, e a senha garanta que a pessoa utilizando este usuário seja realmente quem diz ser.

2.4. **Pentest**

Teste de Intrusão ou Teste de Penetração é um conjunto de métodos utilizados para se avaliar a segurança de sistemas buscando vulnerabilidades no mesmo. Pode-se utilizar das mais variadas táticas para se obter acesso a um sistema, as principais são o *brute force*, engenharia social, análise de portas e negação de serviço. O objetivo geral do *pentest* é encontrar falhas em sistemas e propor contramedidas para saná-las. Este tipo de teste é geralmente executado por hackers.

> Testes de invasão têm por objetivo verificar a resistência de redes, sistemas ou aplicações em relação aos atuais métodos de ataque. Diariamente são descobertas novas falhas nos mais variados sistemas, por isso é de fundamental importância auditorias preventivas, mais especificamente, Testes de Invasão, que podem dar um diagnóstico real sobre a segurança dos ativos em questão. (SOARES, 2010)

[3] Digital Video Recorder (Gravador Digital de Vídeo)

2.5. Endereçamento

Para que dispositivos, como DVRs, possam ser acessados remotamente é necessário que este possua um endereço IP, uma conexão ativa com a *Internet* e, que uma série de requisitos sejam atendidos para viabilizar este acesso.

2.5.1. Endereço IP

Endereço IP é um número de 32 *bits* representado através de 4 octetos no formato decimal, por exemplo, 192.168.0.2 ou 72.30.203.4. Os IPs são dividido em 5 classes conhecidas como classes A, B, C, D e E. Para fins deste estudo separaremos os IPs em dois grandes grupos: a) os privados, utilizados em redes domésticas e empresariais e geralmente não são alcançáveis através da *Internet*, também conhecidos como IPs internos, e b) os IPs públicos, que são endereços alcançáveis através da *Internet* e são utilizados em portais e provedores de serviços *online*.

Os endereços IPs (públicos e privados) são ainda classificados em dois tipos: dinâmicos e estáticos.

Os IPs dinâmicos privados são aqueles que mudam cada vez que um dispositivo se conecta a uma rede ou quando o tempo de concessão[4] é expirado. IPs dinâmicos públicos são fornecidos pelos provedores de *Internet* a cada vez que é realizada uma conexão, como é o caso da banda larga (Speedy®). Caso o assinante se mantenha conectado na internet por dias seu endereço IP será o mesmo por todo esse período de tempo.

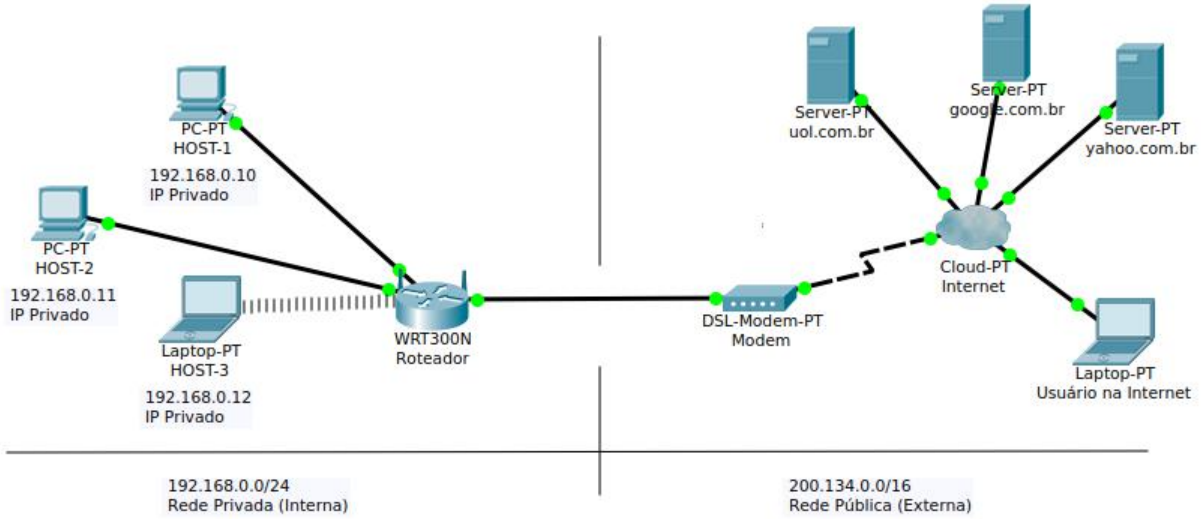
IPs estáticos são endereços que nunca variam, ou seja, são aqueles que são atribuídos a servidores de hospedagem, e-mail, ou à empresas que contratam planos de *Internet* dedicada, assim, não importa quantas vezes o

---

[4] O tempo de concessão (*lease time* ou *lease duration)* controla o tempo que uma máquina pode usar determinado IP. Se após esse tempo o dispositivo não estiver ativo, na próxima vez que se conectar receberá um novo endereço IP (TANENBAUM, 2003, p.351)

modem seja reiniciado ou a conexão refeita, aquele equipamento sempre receberá o mesmo endereço IP. A Figura 1 exemplifica o uso de endereços públicos e privados.

Figura 1 - Rede Privada e Rede Pública



FONTE: PRÓPRIA (2017)

Na Figura 1 a Rede Privada consegue se comunicar diretamente com a Rede Pública, mas o contrário não acontece, pois a rede privada se comunica com a Internet através de um único IP público que normalmente é fornecido pelo provedor de acesso. Para um determinado host[5] da rede interna se comunicar com a Internet é utilizada uma técnica conhecida como NAT (Network Address Translation), que foi concebida com o objetivo de contornar o problema da escassez de endereços IPv4.

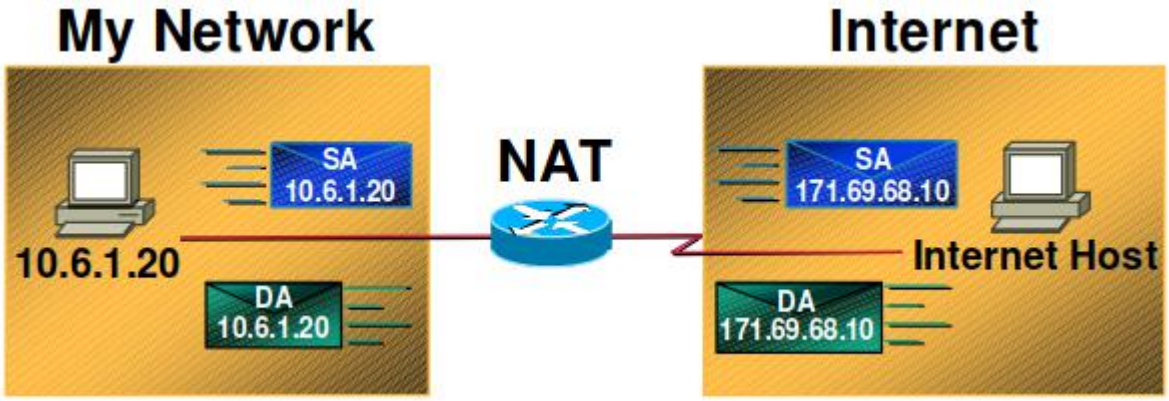
2.5.2. NAT e PAT

“Hoje em dia a idéia em vigor é a de permitir o acesso controlado aos recursos de sua rede interna por usuários da rede externa.” (TORRES, 2001, p. 417).

[5] Por definição, host é qualquer computador ou máquina conectado a uma rede, que conta com número de IP e nome definidos. (VIANA, 2012)

Conforme visto anteriormente, NAT é a técnica utilizada para que um determinado *host* localizado na rede interna consiga se comunicar com a *Internet* e traz consigo o conceito de 1 pra 1, ou seja, 1 *host* na rede interna fazendo uso de 1 IP público ou vice-versa. Desta forma o endereço interno é traduzido para o endereço externo o qual faz o acesso ao site ou serviço na *Internet* (CISCO, 2000). A Figura 2 exemplifica o funcionamento do NAT, onde SA significa Endereço de Origem e DA endereço de destino.

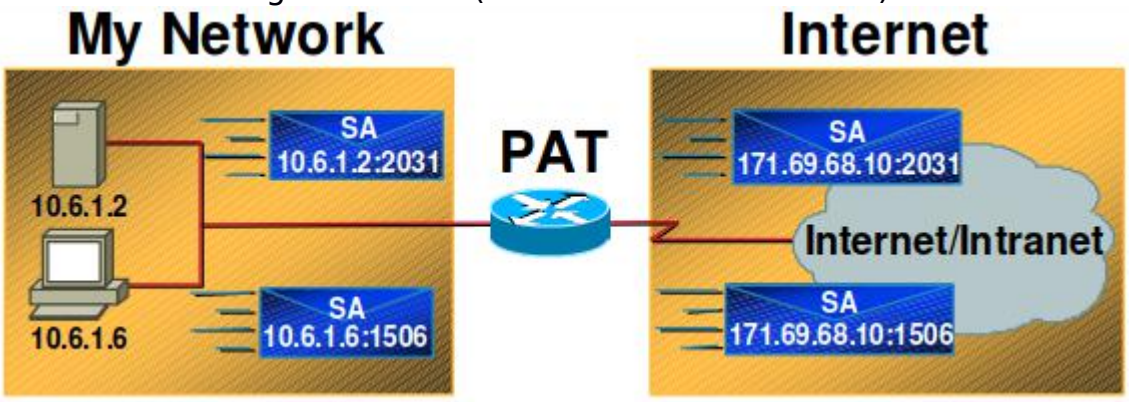
Figura 2 - NAT (Network Address Translation)



FONTE: CISCO (2002)

Já o PAT, ou *Port Address Translation*, extende o uso do NAT sendo possível que vários *hosts* na rede interna acessem a *Internet* através das 65.535 portas de serviços disponíveis. O termo PAT é dificilmente utilizado sendo mais referenciado como NAT.

Figura 3 - PAT (Port Address Translation)



FONTE: CISCO (2002)

É o PAT que torna possível que um grande número de *hosts* em uma rede acessem *sites* e outros serviços na *web* possuindo apenas um endereço IP público. No exemplo da Figura 3, toda requisição é enviada carregando consigo o endereço e porta de origem e destino, quando é devolvida, segue exatamente para a mesma porta garantindo a entrega para o *host* específico. E é através do PAT que conseguimos disponibilizar serviços em uma rede interna para acesso através da *Internet* e neste caso falamos do *Port Forwarding.*

2.5.3. Port Forwarding

Conforme a definição do site PCMagazine:

> Também chamado de "mapeamento de portas", o encaminhamento de portas é direcionar o tráfego do mundo exterior para o servidor apropriado dentro de uma rede local TCP / IP. Os serviços de Internet são identificados por um número de porta padrão; por exemplo, o tráfego da Web usa a porta número 80. Se a rede local hospedar um servidor da Web acessível na Internet pública, o painel de encaminhamento de porta no roteador seria configurado para direcionar os pacotes da Web / HTTP (tráfego da porta 80) para o IP endereço do servidor da Web na rede local (LAN)

Desta forma, para que seja possível o acesso à um recurso da rede local através da *Internet*, é necessária a configuração do roteador para que ele direcione o tráfego para o *host* e porta corretos. A Figura 4 mostra o painel de configuração de direcionamento de portas.

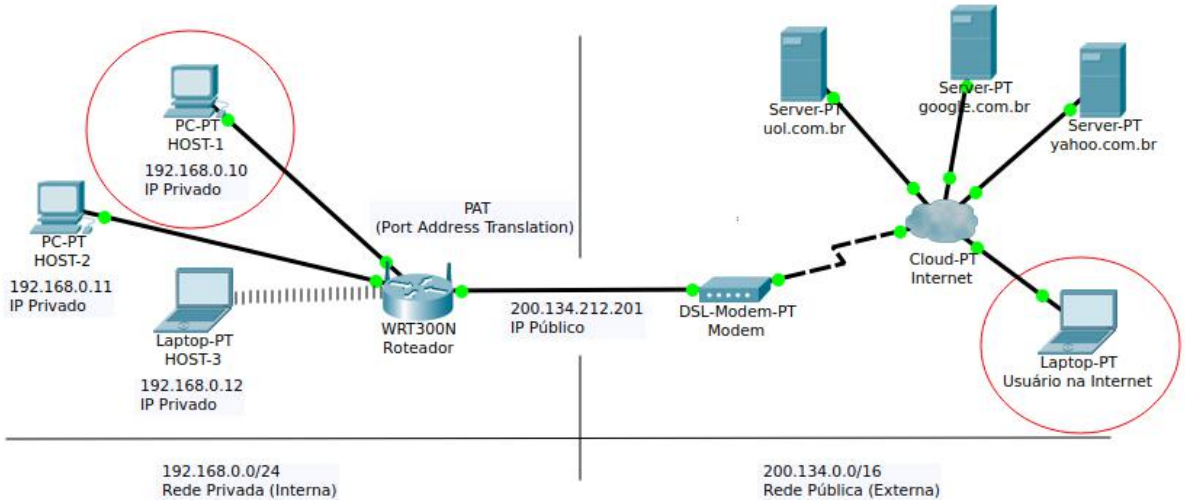
Figura 4 - Port Forwarding (Roteador TP-Link)

FONTE: TP-LINK (2017)

Usando a Figura 4 como referência e supondo que o IP público deste roteador é 200.134.212.201, quando um acesso originar da *Internet* para este IP na porta 8080, este tráfego será redirecionado para o *host*

192.168.0.10 na porta 80. Utilizando a Figura 5 como exemplo, se o Usuário na *Internet* digitar em seu navegador http://200.134.212.201:8080, será carregada a página *web* hospedada no *host* 192.168.0.10 que está dentro de uma rede privada.

Figura 5 - PAT - Tradução de endereço para vários hosts



FONTE: PRÓPRIA (2017)

2.5.4. **DNS**

Como visto, cada dispositivo em uma rede TCP/IP deve possuir um endereço IP para que este seja alcançável através desta rede. Contudo, endereços IP não são fáceis de serem decorados e para resolver este problema “foi criado o sistema DNS (Sistema de nome de domínio) que permite dar nome a endereços IP, facilitando a localização de máquinas por nós, humanos” (Torres, 2001, p. 114)

O DNS permite que consigamos acessar serviços (sites, arquivos, etc) sem termos que nos preocupar em saber o endereço IP de onde este serviço está hospedado, bastando apenas conhecer o nome do domínio do serviço que desejamos nos conectar. Assim, quando digitamos em nosso navegador um endereço, por exemplo www.uol.com.br, o DNS informa que o site www.uol.com.br está hospedado no endereço IP 200.147.67.142. De formal simples e didática o Quadro 1 ilustra o funcionamento de um sistema DNS.

Quadro 1 - Tabela de DNS

Nome de Domínio	Endereço IP
www.uol.com.br	200.147.67.142
www.google.com.br	172.217.29.163
www.yahoo.com.br	72.30.203.4

FONTE: PRÓPRIA (2017)

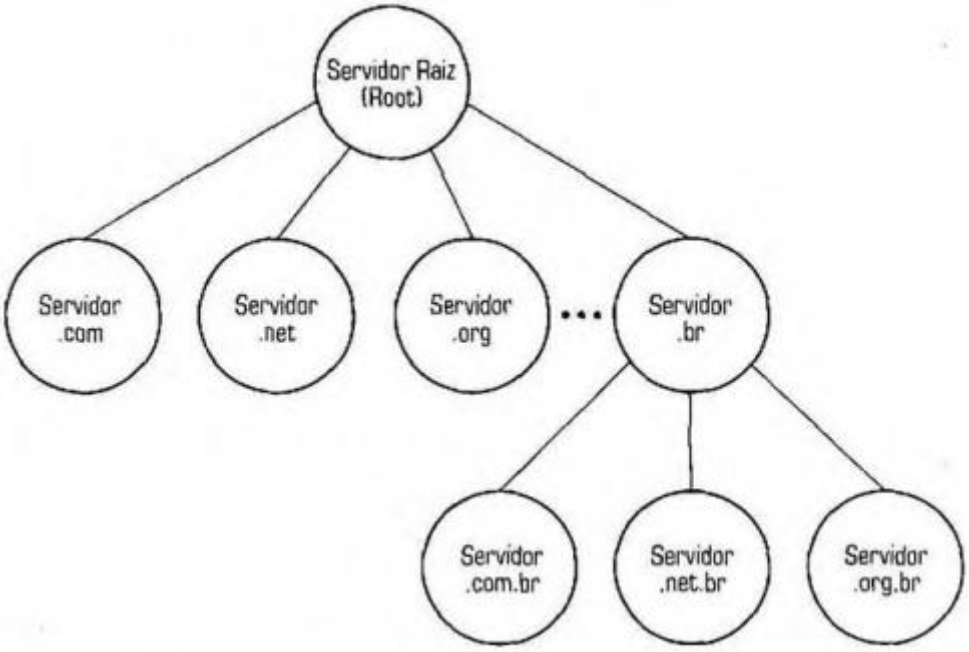
2.5.4.1. Servidores DNS

Conforme visto, o DNS é o responsável por converter endereços nominais em IPs, mas onde está armazenada esta tabela?

> Sem o uso de servidores DNS, cada máquina conectada na *internet* teria de ter uma tabela contendo todos os endereços IP e os nomes das máquinas, o que atualmente é impossível, já que existem milhões de endereços na *internet*. Como a idéia da *internet* é ser uma rede gigantesca – e ela foi criada justamente para isso -, o sistema de nomes foi criado de forma a acomodar o crescimento da rede. Para isso, o sistema de nomes utilizado possui uma estrutura hierárquica. (Torres, 2001, p. 114)

Logo, existem servidores espalhados de forma hierárquica pela *Internet* que são responsáveis pela “tradução” de nomes de domínio em endereços IP. Quando estamos conectados na *Internet* possuímos geralmente o endereço IP de um servidor DNS Primário, que é o primeiro lugar onde serão pesquisados os nomes de domínio durante a navegação. A partir desse ponto, caso o nosso servidor primário não conheça o endereço requisitado, ele apontará para um próximo servidor onde a busca será reiniciada. A hierarquia de servidores DNS pode ser observada na Figura 6.

Figura 6 - Hierarquia de servidores DNS



FONTE: TORRES (2001, P. 114)

> Na Figura 6 vemos um exemplo de como funciona essa estrutura. Um servidor .com, por exemplo, é responsável por todos os endereços terminados em .com, assim como um servidor .com.br é responsável por todos os endereços terminados em .com.br. (TORRES, 2001, p. 115)

Assim, ao digitarmos um endereço no navegador, por exemplo www.uol.com.br, a busca por esse endereço será iniciada no servidor .br, o qual apontará para o servidor .com.br e, neste, será encontrado o endereço digitado e, assim, carregada a página. Caso o endereço não seja encontrado no servidor .com.br será retornado o conhecido erro 404 – página não encontrada.

2.5.4.2. Serviço de DDNS

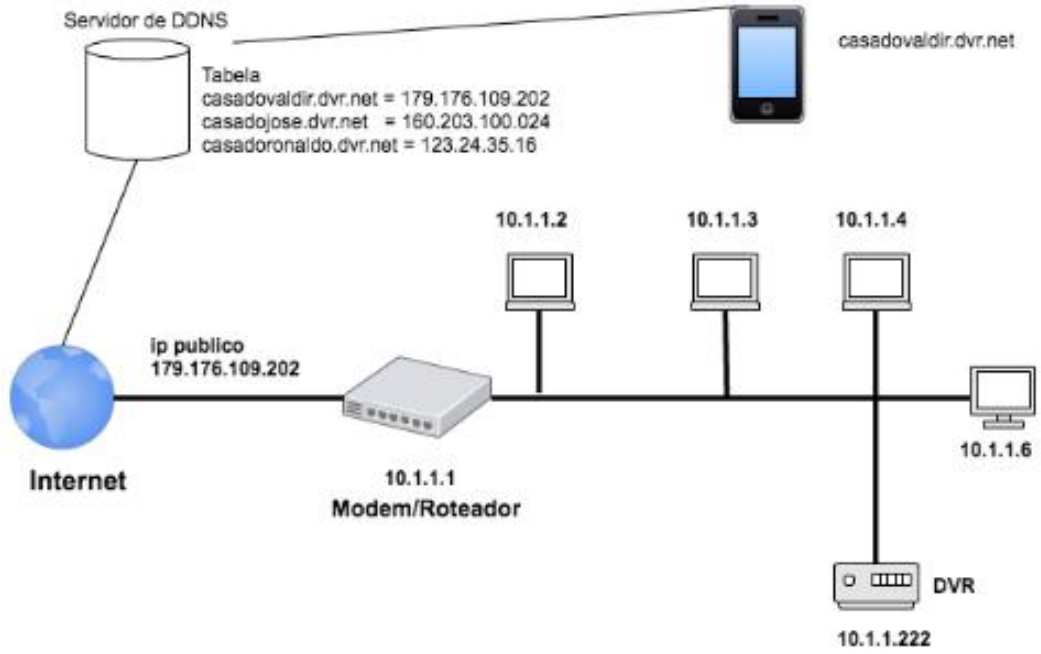
O *Dynamic Domain Name System*, assim como o DNS, é um sistema para tradução de nomes de domínio para endereços IP. A grande diferença é que este sistema se destina a conexões que possuem IPs dinâmicos, assim, clientes que possuem este tipo de IP, podem ter serviços disponibilizados na *Internet* através de um nome de domínio mesmo que seu IP se altere frequentemente.

> É fácil para as grandes empresas configurar nomes de domínio como o Facebook.com, porque o endereço do servidor *web* é estático (uma vez que eles possuem o endereço IP ele não se altera). Pessoas com conexões residenciais obtêm um endereço IP dinamicamente atribuído[...] Assim [...]o endereço que você possui hoje não é o endereço que você pode ter na semana que vem. Felizmente, os provedores DDNS tornam simples para atribuir um nome memorável ao endereço IP de sua casa porque eles o atualizam automaticamente à medida que seu endereço IP muda ao longo do tempo. (FITZPATRICK, 2016)

Conforme Rosa (2016), DDNS de forma geral nada mais é que um servidor na *Internet* que armazena em uma tabela de seu banco de dados duas informações: 1) O IP público atual do cliente (179.176.109.202) 2) Seu endereço na internet (algo como casadovaldir.dvr.net)

Esse sistema funciona basicamente através de um recurso, presente na maioria dos roteadores, que atualiza o servidor DDNS toda vez que o endereço IP do cliente é alterado (mesmo que isso ocorra várias vezes no mesmo dia).

Figura 7 - Funcionamento do DDNS



FONTE: ROSA (2016)

Na Figura 7 perceba que o servidor DDNS armazena e relaciona em uma tabela o IP público atual com o endereço DNS de cada cliente. De forma que se for necessário acessar, por exemplo, o seu DVR por uma outra rede, você precisará conhecer apenas o seu endereço DNS (casadovaldir.dvr.net). Ao digitar seu endereço (casadovaldir.dvr.net) em seu smartphone, o servidor de DDNS faz o desvio para IP público atual (179.176.109.202) (ROSA, 2016).

Tomando como exemplo a Figura 7, o Modem/Roteador envia, a cada nova conexão, o endereço IP atual ao servidor DDNS. Existem serviços dedicados a prover este tipo de solução, os mais populares são: DynDNS (www.dyndns.org) e No-IP (www.no-ip.com).

Do mesmo modo, fabricantes de dispositivos de monitoramento, como é o caso da Tecvoz e da Alive Eletronics, disponibilizam um serviço de DDNS para os consumidores que possuem IPs dinâmicos em sua residência ou empresa. Assim, os clientes podem escolher um nome dentro do domínio DDNS da empresa para realizar o acesso de qualquer lugar, como por exemplo, unip.tecvozddns.com.br ou jose.aliveddns.com.br. Desta maneira, quando o usuário acessar esse endereço, ele será direcionado para o IP de onde o sistema de câmeras está instalado. O Quadro 2 ilustra, com dados fictícios, o funcionamento de um DDNS.

Quadro 2 - Tabela DDNS

DNS	IP
jose.tecvozddns.com.br	189.238.21.2
rihappy.tecvozddns.com.br	200.43.123.34
maria.tecvozddns.com.br	176.34.124.66

FONTE: PRÓPRIA (2017)

Neste caso, o servidor DDNS atua como um centralizador direcionando um nome de domínio para um IP que pode estar localizado em qualquer lugar do mundo.

Agora que entendemos o conceito de IP, Porta e DNS sabemos onde iniciar a busca por equipamentos de segurança.

2.6. Encontrando dispositivos

Saber qual é o servidor DDNS responsável pelo endereçamento de DVRs é o primeiro passo para que seja possível descobrir o endereço DNS dos mesmos. Como os subdomínios dentro de um servidor DDNS não são divulgados ou indexados por mecanismos de busca como o Google® é necessária a utilização de alguma técnica para encontrar um dado endereço, por exemplo cazuza.tecvozddns.com.br (que não sabemos se de fato existe). Neste ponto utilizamos duas das técnicas já citadas para tornar esta tarefa menos onerosa: Automação, executando a tarefa por meio de um software e, Ação à distância, o software fará os testes em um servidor DDNS localizado geograficamente distante do computador de origem.

Sabendo onde estão centralizados os endereços dos dispositivos podemos iniciar a busca. A tática utilizada neste estudo foi a de DNS Brute Force, que é uma forma de Enumeração de DNS Automatizada (MCCLURE, 2009, p. 91) que conceitualmente consiste em utilizar uma lista com os nomes de subdomínio mais comuns (mail, www, www3, webmail, etc) e verificar se os mesmos estão ativos no domínio principal.

Como domínios DDNS de câmeras de segurança geralmente não utilizam nomes comuns para referenciar dispositivos, foi escrito um software (APÊNDICE 1) para tentar todas as possibilidades de nomes de domínios dentro de um intervalo e, armazenar aqueles que possuem uma interface web disponível. Como o DDNS escaneado pertence a uma fabricante de dispositivos de monitoramento a grande maioria dos dispositivos encontrados serão deste tipo. “Escanear é o equivalente a bater nas paredes para encontrar todas as portas e janelas” (MCCLURE, 2009, p. 44).

2.6.1. Brute Force

Assim, a primeira técnica utilizada para a descoberta de nomes de domínios, que correspondem à dispositivos de monitoramento, foi o *Brute*

Force ou Ataque por Força Bruta. Este método consiste na adivinhação, manual ou automatizada, geralmente de senhas, através da tentativa e erro de todas as combinações possíveis de um conjunto de caracteres. É um método que exige um tempo considerável para ser executado uma vez que existem milhões ou bilhões de possibilidades dependendo do conjunto de caracteres utilizados e do tamanho da palavra final. “Por exemplo, para se gerar senhas utilizando as 26 letras do alfabeto com um limite de 7 caracteres, temos mais de 8 bilhões de possibilidades.” (MCCLURE, 2009, p. 54).

Suponhamos que um sistema possui uma senha de 6 dígitos numéricos para acesso, na primeira tentativa usaríamos a senha 1, em seguida 2, 3, e assim sucessivamente até que a correta seja encontrada. No pior dos casos a correta seria 999999 e dependendo do sistema essa senha levaria dias para ser descoberta. Usaremos esta técnica para a descoberta de nomes de domínios.

O *software* é executado dentro de um *loop* onde são definidos os caracteres que serão utilizados para compor o nome do domínio, bem como o tamanho inicial e final destes nomes. Em nosso caso foram utilizadas apenas as 26 letras do nosso alfabeto e o intervalo de 1 a 5 caracteres, o que corresponde à 11.881.376 combinações de nomes de domínio.

Assim que o programa é iniciado a primeira combinação é a.ddns.com.br[6], ou seja, a letra atual “a” concatenada com o domínio DDNS alvo “.ddns.com.br”. Neste ponto é verificado se existe um IP associado a este nome utilizando-se um método específico da liguagem utilizada. Assim, se existir um endereço IP para o nosso nome de domínio, significa que o nosso destino está ativo.

[6] O domínio ddns.com.br foi utilizado apenas para fins didáticos e visando ocultar o domínio real que foi testado.

Figura 8 - Execução de Brute Force em DDNS

FONTE: PRÓPRIA (2017)

Conforme mostrado na Figura 8, existe um IP associado ao domínio m.ddns.com.br, o que significa que serviços podem estar ativos neste endereço. Para descobrirmos quais serviços estão disponíveis neste endereço utilizaremos um port scanner.

2.6.2. Port Scanner

Conforme McClure (2009, p. 54) *“Port scanning* é o processo de enviar pacotes para portas TCP e UDP no sistema alvo para determinar quais serviços estão em execução“.

Assim, *Port Scanners* ou varredores de porta, são *softwares* utilizados para descobrir portas (TCP e UDP) disponíveis em uma rede. Para a realização de buscas deve-se fornecer como parametros ao programa uma rede de computadores (192.168.0.0/24) ou um único *host* (192.168.0.10) e, opcionalmente, uma faixa de portas a serem escaneadas (Ex: 25 à 80), para que o programa faça a varredura e retorne os *hosts* que estão disponíveis e quais portas estão acessíveis.

Programas desta categoria são geralmente utilizados por *hackers* e *crackers* buscando vulnerabilidades em redes de computadores, sendo o Nmap o mais conhecido.

Como um único IP público pode fornecer uma grande variedade de serviços através do recurso de *Port Forwarding* (RFC6886, 2013), é necessária a detecção de quais portas estão disponibilizando serviços em determinado IP, assim fazemos uso do *portscan* para obter estas portas.

2.6.3. Nmap

O Nmap, ou *Network Mapper*, é um *Port Scanner* gratuito e de código aberto utilizado para mapeamento de serviços disponíveis em uma rede. Ele é um software CUI (Command User Interface), ou seja, é utilizado através da linha de comandos do sistema operacional, mas possui também uma GUI (Graphical User Interface) chamada ZenMap que torna seu uso mais interativo.

Para a utilização simples do Nmap, deve ser fornecido como argumento ao programa a Rede, o endereço IP, ou o nome de domínio do *host* que se deseja escanear. A Figura 9 mostra um exemplo de uso do nmap em modo de linha de comando.

> O Nmap oferece opções para especificar quais portas são escaneadas e se a ordem de escaneamento é aleatória ou sequencial. Por padrão, o Nmap escaneia todas as portas até, e incluindo, 1024, bem como portas com numeração alta listadas no arquivo the `nmap-services` para o(s) protocolo(s) escaneados. (NMAP.ORG)

Figura 9 - Listando portas com Nmap

```
➜ tools git:(master) ✗ nmap -Pn m.[redacted].com.br

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-28 22:12 -02
Nmap scan report for m.[redacted].com.br (187.1[redacted]8.8)
Host is up (0.038s latency).
rDNS record for 187.1[redacted]8.8: 187.1[redacted]8.8.static.gvt.net.br
Not shown: 930 filtered ports, 65 closed ports
PORT     STATE SERVICE
554/tcp  open  rtsp
3389/tcp open  ms-wbt-server
9000/tcp open  cslistener
9001/tcp open  tor-orport
9009/tcp open  pichat

Nmap done: 1 IP address (1 host up) scanned in 4.61 seconds
```

FONTE: PRÓPRIA (2017)

Como podemos observar o domínio m.ddns.com.br possui 4 portas abertas (open), sendo elas 554, 9000, 9001 e 9009. Agora precisamos saber, em qual/quais destas está o serviço HTTP que é o meio pelo qual acessamos a interface de um DVR.

2.7. Encontrando a porta certa

Sabendo que dispositivos DVR oferecem um servidor HTTP interno para que seja possível a visualização das imagens através de um navegador, verificamos em cada uma das portas abertas se existe um servidor HTTP disponível. Para esta verificação foram realizados métodos da liguagem PHP e que podem ser conferidos no APÊNDICE 1

Se existir um servidor disponível, é registrado o endereço de domínio e a porta em um arquivo de texto para posterior análise, compondo assim um banco de dados de dispositivos ativos.

Para fins deste estudo o programa foi configurado para testar todas as possibilidades de nomes de domínios iniciando em “a” e finalizando em “zzzzz” e testar todas as portas padrão abertas para cada um dos domínios encontrados.

Neste intervalo foram encontrados 1654 endereços com um servidor HTTP ativo e a busca levou aproximadamente 435 horas para ser concluída.

Figura 10 - Buscando um servidor HTTP

FONTE: PRÓPRIA (2017)

A Figura 10 mostra os testes de portas sendo realizados. No domínio m.ddns.com.br o programa encontra as portas 9000, 9001 e 9009 abertas, mas apenas a 9001 responde como um serviço HTTP. Esta resposta é analizada através de uma requisição HTTP enviada ao domínio e porta, se a resposta for um texto no formato HTML este é registrado. Assim, o arquivo de texto discovered-http.txt é populado com a data e hora de registro, o IP do domínio, o domínio DNS e porta do dispositivo como exibido na Figura 11.

Figura 11 - conteúdo do arquivo discoved-http.txt

FONTE: PRÓPRIA (2017)

Até este ponto sabemos o endereço de cada dispositivo que iremos fazer a análise de segurança. Agora precisamos testar as senhas padrão para provar que a segurança de determinado dispositivo está comprometida.

2.8. Tentando o acesso com Senhas Padrão

Na maioria dos sistemas embarcados[7], o primeiro acesso é realizado utilizando as credenciais padrão do dispositivo. Cada fabricante adota um padrão próprio para esta autenticação conforme mostra o Quadro 3.

Quadro 3 - Senhas padrão de dispositivos

Tipo de Dispositivo	Fabricante	Usuário	Senha
Roteador	D-Link	admin	(em branco)
Roteador	TP-Link	admin	admin
DVR	Tecvoz	admin	1
DVR	Alive Eletronics	admin	(em branco)

FONTE: PRÓPRIA (2017)

Como esta autenticação é padrão para qualquer dispositivo desta marca, é altamente recomendado que este usuário e senha sejam alterados assim que o primeiro acesso for realizado, o que muitas vezes não acontece e, alguns usuários, ainda utilizam senhas como 123456 para acesso a seus dispositivos.

Manter credenciais padrão em dispositivos conectados à *Internet* equivale a trancar sua casa ou carro e deixar a chave na porta. “Uma senha mal elaborada, fácil de ser decifrada, pode ser obtida por sujeitos mal-intencionados, e uma vez que autenticado como outra pessoa, obter informações privilegiadas e desferir ataques sem ser identificados.” (ABREU, 2011, p. 16)

Assim, em sistemas de monitoramento, se um indivíduo mal-intencionado ganha acesso às imagens de segurança, este pode usar as mesmas para realizar furtos ou outros crimes, uma vez que saberá exatamente a hora que pode agir.

Os computadores se destacam pela possibilidade de execução de tarefas de forma massiva e repetitiva. Assim, nosso atacante, poderia rackear sistemas enquanto dorme. (SCHNEIER, 2000, p. 18). Desta forma

[7] Sistema embarcado é aquele que dá capacidade computacional a circuitos integrados, equipamentos ou sistemas (CUNHA, 2007)

podemos executar tentativas de acesso de forma automatizada enquanto realizamos outras tarefas.

Durante os testes foram encontrados dezenas de modelos de DVRs e Câmeras IP, e embora usem usuário e senha para o acesso, alguns deles exigem a instalação de um *plugin*[8], por onde é realizada a autenticação do usuário.

Assim foi foi criado o script logintest.php, e não disponibilizado neste trabalho, o qual faz a separação dos modelos de DVR em dois grupos: os que aceitam autenticação via HTTP e os que exigem a instalação de *plugin* para esta autenticação. Para aqueles que aceitam autenticação HTTP já é realizada nesse passo a tentativa de acesso com as senhas padrão.

Este programa recebe como entrada o arquivo de saída do APÊNDICE 1 (discovered-http.txt) e retorna 4 arquivos: 1) r-manual.txt o qual contém a quantidade de URLs que exigem instalação de *plugin*. 2) r-encontrados.txt que contém a quantidade de URLs dos dispositivos que foram acessíveis através de uma senha padrão. 3) r-nao-encontrados.txt neste arquivo está a quantidade de dispositivos que não estavam acessíveis através de usuário e senha padrão. 4) r-nao-verificados.txt contendo a quantidade de URLs que não foram verificadas por algum motivo. Ex: URL não encontrada.

Os testes foram executados enviando-se o usuário e senha padrão para uma determinada URL em cada um dos dispositivos encontrados. Esta URL retorna o valor 200 (Figura 12) quando o dispositivo aceita as credenciais padrão enviadas e 401 (Figura 13) quando não aceita.

[8] Um plugin é um complemento de software que é instalado em um programa, permitindo que ele execute recursos adicionais. Por exemplo, os navegadores da Internet permitem aos usuários instalar plug-ins no navegador para dar a ele recursos não são encontrados na instalação padrão.(EMBERTON, 2017)

Figura 12 - Resposta XML em caso de sucesso na autenticação

```
<?xml version="1.0" encoding="UTF-8" ?>
<userCheck>
<statusValue>200</statusValue>
<statusString>OK</statusString>
</userCheck>
```

FONTE: PRÓPRIA (2017)

Figura 13 - Resposta XML em caso de falha na autenticação

```
<?xml version="1.0" encoding="UTF-8" ?>
<userCheck>
<statusValue>401</statusValue>
<statusString>Unauthorized</statusString>
</userCheck>
```

FONTE: PRÓPRIA (2017)

Para os testes foram utilizadas somente as credenciais exibidas no Quadro 4.

Quadro 4 - Senhas utilizadas nos testes

Usuário	Senha
admin	1
admin	admin
admin	123456
admin	12345
admin	1234
admin	12
admin	(em branco)

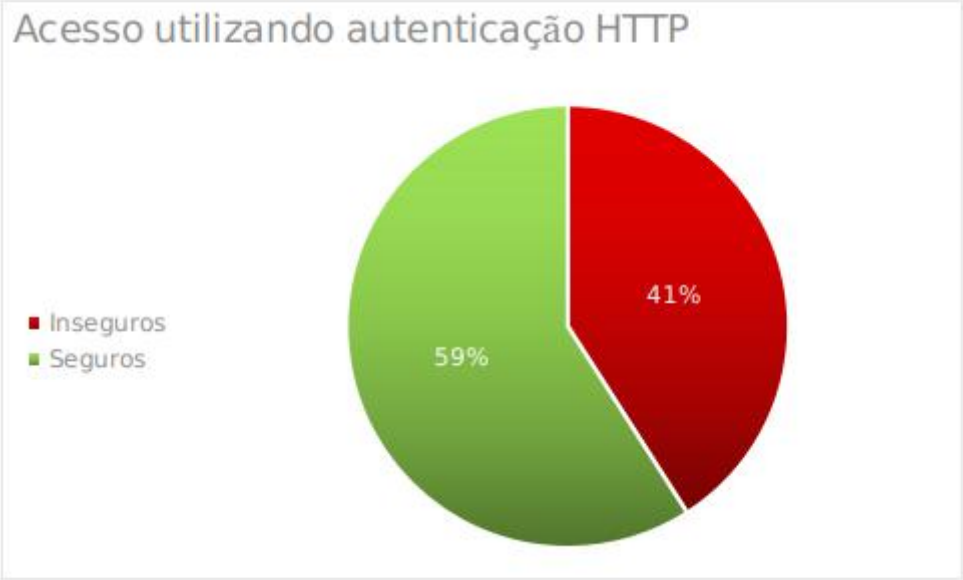
FONTE: PRÓPRIA (2017)

2.9. Resultados e Análise Experimental

Ao término da execução dos testes de login foram encontrados 712 dispositivos e analisados 391 (os equipamentos que utilizavam *plugin* para autenticação foram removidos dos testes), e destes, 160 foram classificados como inseguros, ou seja, estavam acessíveis através de uma das senhas exibidas no . 41% dos dispositivos aceitaram uma das

credenciais informadas, conforme a Figura 14 e destes, 90% utilizavam credenciais padrão do dispositivo, ilustrado na Figura 15.

Figura 14 - Dispositivos utilizando autenticação HTTP



FONTE: PRÓPRIA (2017)

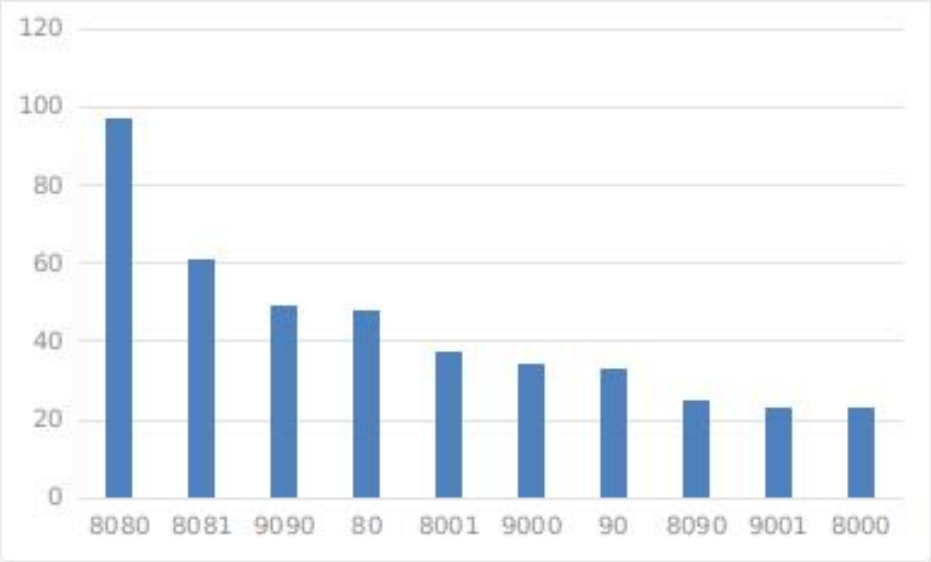
Figura 15 - Senhas utilizadas

FONTE: PRÓPRIA (2017)

2.9.1. Portas mais utilizadas

Durante a análise dos resultados foi verificado o padrão de portas utilizadas para o acesso aos dispositivos de monitoramento. Para os 712 dispositivos encontrados apenas um total de 91 portas foram utilizadas. Desta forma, ao executar o comando nmap para varrer as portas dos dispositivos, pode-se especificar as portas mais relevantes buscando um melhor desempenho do processo. A Figura 16 mostra as 10 portas mais utilizadas para acesso de acordo com os dados coletados.

Figura 16 - Portas mais utilizadas



FONTE: PRÓPRIA (2017)

2.10. Por que não damos importância para a segurança digital?

Diferentemente da vida real, no mundo digital acreditamos que roubos ou invasões são coisas de filmes e nunca acontecerão conosco. Um sistema configurado por uma pessoa sem malícias com relação à segurança pode representar riscos inestimáveis à pessoas e empresas. Acreditamos que ninguém descobrirá o endereço IP ou o nome de domínio (DDNS) onde nosso serviço é disponibilizado. Assim, não tomamos o devido cuidado em alterar a porta de entrada ou escolher uma senha segura.

> Com freqüência, a segurança é apenas uma ilusão, que às vezes fica pior ainda quando entram em jogo a credulidade, a inocência ou a ignorância. O cientista mais respeitado do mundo no século XX, Albert Einstein, disse: "Apenas duas coisas são infinitas: o universo e a estupidez humana, e eu não tenho certeza se isso é verdadeiro sobre o primeiro".(MITNICK, 2003, p. 3)

Como uma medida para reforçar a segurança, uma boa prática de *hardening*[9] seria configurar o dispositivo para usar portas não padrão para acesso, evitando assim que o mesmo seja encontrado através do *scan* padrão do *nmap*. Ao invés de ser usada uma porta 8080, 9000 ou 7070, a melhor prática seria a utilização de portas de acesso acima de 50000, assim, para que esta porta seja encontrada, o *nmap* deve ser configurado para que faça o *scan* em todas as 65.535 portas existentes, o que torna o processo demorado e inviável para o atacante.

[9] Hardening é uma técnica usada para mapear ameaças e depois executar possíveis correções nos sistemas, preparando-os para determinadas tentativas de ataques ou violação na segurança da informação (FACINA, 2009)

3. CONCLUSÃO

Este estudo demonstrou, com base nos endereços encontrados, que um número considerável de dispositivos de monitoramento, tanto aqueles instalados por empresas de segurança qualificadas como os instalados pelos próprios usuários, estão acessíveis através de senhas padrão dos fabricantes ou senhas inseguras. Segundo o empresário Domingos Vasques da Silva[10] “Em toda a instalação de sistemas de monitoramento o usuário é instruído a alterar a senha padrão, mas este geralmente prefere senhas fáceis de lembrar, como nomes, datas especiais e até mesmo sequências como 123456”.

Para buscarmos uma solução para esta prática devemos entender porquê as pessoas continuam a usar senhas inseguras. No artigo “Por que as pessoas ainda usam senhas péssimas na internet?” Hamann (2017) enumera os principais motivos do uso de senhas inseguras e que resumem-se em a) Pouca preocupação com segurança, b) Fácil memorização e c) Os próprios serviços permitem.

Muito tem se investido na questão da conscientização dos usuários no que diz respeito à segurança da informação e, mesmo após os ciberataques noticiados mundialmente em maio de 2017, o que deveria motivar melhores práticas de prevenção, as pessoas ainda não se preocupam utilizando a frase “quem é que vai querer saber do que eu tenho na minha conta de e-mail?” (HAMANN, 2017). O sentimento de segurança é o mesmo que nos torna vulneráveis, uma vez que quanto mais seguros nos sentimos, mais baixamos a guarda e tendemos a ter comportamentos inseguros.

Neste cenário, onde os usuários não se preocupam com sua própria segurança, cabe aos provedores de serviços e fabricantes de dispositivos criar medidas para o descaso com senhas de acesso. Uma boa alternativa, que já é adotada por alguns fabricantes, é utilizar como senha padrão do

---

[10] Empresário proprietário da empresa de segurança Arthur Seg. contatoarturseg@hotmail.com

dispositivo os últimos 6 dígitos do endereço MAC[11] ou um PIN[12], assim não é possível simplesmente adivinhar uma senha. Além destas medidas, a implantação de uma política interna de senhas poderia garantir que o usuário não utilizasse senhas inseguras em seus dispositivos.

Exigindo-se também que seja informado um e-mail do proprietário do dispositivo e do responsável técnico seria uma forma de garantir que, em qualquer atividade suspeita ou notificação de atualização, os interessados sejam notificados.

Assim, com apenas algumas adequações no software embarcado, os próprios fabricantes garantem que os usuários não utilizem senhas que podem comprometer sua própria segurança.

> “Agora mais do que nunca devemos aprender a parar de ser otimistas e nos tornarmos mais conscientes das técnicas que estão sendo usadas por aqueles que tentam atacar a confidencialidade, integridade e disponibilidade das informações dos nossos sistemas e redes de computadores. Nós acostumamo-nos a aceitar a necessidade da direção segura; agora está na hora de aceitar e aprender a prática da computação defensiva.” (MITNICK, 2003, p. 8)

[11] Sigla de Media Access Control, uma sequência única que identifica cada placa de rede.
[12] Personal Identification Number, um número aleatório gerado no processo de fabricação.

4. TRABALHOS FUTUROS

Mostrar que não somente dispositivos de monitoramento estão expostos na Internet como também roteadores, equipamentos VoIP, Relógios Ponto, sistemas de gerenciamento de provedores de internet, etc.

Analisar o nível de preocupação das empresas com relação à segurança da informação.

5. REFERÊNCIAS

BORBA, Antonio. **Site alerta para a vulnerabilidade de câmeras de segurança.** Disponível em <https://www.magicwebdesign.com.br/blog/internet/site-alerta-para-vulnerabilidade-de-cameras-de-seguranca/> Acesso em 15 out. 2017.

BRASIL. Lei 12.737, de 30 de novembro de 2012. **Dispõe sobre a tipificação criminal de delitos informáticos**; altera o Decreto-Lei no 2.848, de 7 de dezembro de 1940 - Código Penal; e dá outras providências.

CHESHIRE, Stuart; KROCHMAL, Marc. **NAT port mapping protocol**. Disponível em <https://tools.ietf.org/html/rfc6886>. Acesso em 30 abr. 2017.

CISCO. **Introduction to Network Address Translation** Disponível em <https://www.cisco.com/networkers/nw00/pres/2211.pdf>. Acesso em 27 nov. 2017

CUNHA, Alessandro F. **O que são sistemas embarcados**. Saber Eletrônica, 2007. v. 43, n. 414, p. 1-6.

EMBERTON, Nathan. **Plugin**. Disponível em <https://www.computerhope.com/jargon/p/plugin.htm>. Acesso em 15 out. 2017.

FACINA, André Luiz. **Hardening no OpenBSD**. Disponível em <https://www.vivaolinux.com.br/dica/Hardening-no-OpenBSD/>. Acesso em 15 out. 2017

FERREIRA, Nicholas. O Guia do Hacker. Disponível em <http://www.guiadohacker.com.br/O_Guia_do_Hacker_1_edicao.pdf>. Acesso em 20 jun. 2017

FITZPATRICK, Jason. **How To Easily Access Your Home Network From Anywhere With Dynamic DNS**. Disponível em <https://www.howtogeek.com/66438/how-to-easily-access-your-home-network-from-anywhere-with-ddns/> Acesso em 13 out. 2017.

G1. **Ciberataques em larga escala atingem empresas no mundo e afetam Brasil**. Disponível em <http://g1.globo.com/tecnologia/noticia/hospitais-publicos-na-inglaterra-sao-alvo-cyber-ataques-em-larga-escala.ghtml>. Acesso em 20 mai. 2017.

HAMANN, Renan. **Por que as pessoas ainda usam senhas péssimas na internet?** Disponível em <https://www.tecmundo.com.br/seguranca/113490-por-que-pessoas-ainda-usam-senhas-pessimas-internet.htm>. Acesso em 2 fev. 2017.

MARTINS, Claudemir. **Como Hackear câmeras de segurança e DVRs**. Disponível em <http://aprendacftv.com/10-segredos-dos-hackers-que-invadem-cameras-de-seguranca-e-dvrs/>. Acesso em 15 out. 2017

MCCLURE, Stuart; KURTZ, George; SCAMBRAY, Joel. **Hacking exposed 6:** network security secrets & solutions. 6ª ed. New York:McGraw-Hill, 2009. 720 p.

MITNICK, Kevin D.; SIMON, Willian L. **Mitnick:** a arte de enganar**.** São Paulo: Pearson, 2003.

NETWORK Mapper: Introduction. Disponível em <https://nmap.org/#intro>. Acesso em 30 abr. 2017.

OLIVEIRA, Paulo. **Protocolo TCP e UDP**. Disponível em <http://kretcheu.com.br/videos/protocolos-tcp-e-udp/> Acesso em 12 de out. 2017.

RIBEIRO, Beno. **O que é CFTV e sua História**. Disponível em <http://grupofortseg.net/o-que-e-cftv-e-sua-historia/>. Acesso em 30 abr. 2017.

ROSA, Marcio. **O Que é DDNS**. Disponível em <http://serenarseguranca.com.br/o-que-e-ddns/>. Acesso em 1 out. 2017.

SCHNEIER, Bruce. **Secrets & Lies: Digital Security in a Networked World**. Indianapolis:Wiley, 2004.

SOARES, Rafael. **Auditoria Teste de Invasão(Pentest) – Planejamento, Preparação e Execução**. Disponível em <https://seginfo.com.br/2010/09/07/auditoria-teste-de-invasaopentest-planejamento-preparacao-e-execucao-2/>. Acesso em 14 out. 2017.

TANENBAUM, A. S. **Redes de Computadores** – 4ª Ed. Tradução Vandenberg D. de Souza. Campus, 2003.

TARCIO, Paulo. **Ferramentas para pentest: nmap**. Disponível em <http://www.mundodoshackers.com.br/ferramentas-para-pentest-nmap>. Acesso em 30 abr. 2017.

TORRES, Gabriel. **Redes de Computadores:** Curso Completo. Rio de Janeiro:Axcel, 2001.

WEAVER, Matthew. **UK moves to shut down Russian hackers streaming live British webcam footage**. Disponível em <https://www.theguardian.com/technology/2014/nov/20/webcam-hackers-watching-you-watchdog-warns>. Acesso em 15 out. 2017

PCMagazine. **Definition of: port forwarding.** Disponível em <http://www.pcmag.com/encyclopedia_term/0,1237,t=port+forwarding&i=49509,00.asp>. Acesso em 27 nov. 2017

6. APÊNDICE 1 - LOCALIZADOR DE DOMÍNIOS ATIVOS

```
1. <?php
2.
3.  /* Script para executar brute force para descoberta de
4.   * subdomínios ativos em serviços de DDNs
5.   * e encontrar portas HTTP ativas nestes domínios
6.   *
7.   * Autor: Tieferson Leandro Domingos <tiefersond@yahoo.com.br>
8.   * Criação: 07/05/2016     Última Modificação: 13/10/2017
9.   *
10.   * TODO: Utilizar o nmap com a opção -oX, a qual gera uma saída XML dos
11.   * resultados. Ex: nmap -Pn -oX - xyz.ddns.com.br > saida.xml
12.   * Acredito que o interessante seja criar um novo script
13.   * e otimizá-lo com os resultados encontrados
14.   *
15.   */
16.
17.  //Domínio alvo do brute force. Ex: dynns.com, ddns.net
18.  $dominioAlvo="ddns.com.br";
19.
20.  //Caracteres utilizados no bruteforce
21.  $chr="abcdefghijklmnopqrstuvwxyz";
22.
23.  // Tamanho máximo da string. Quando alcançar esse número o script pára.
24.  $maxstrlen=5;
25.
26.  // Portas removidas do script por causarem interrupção do mesmo
27.  $portasDescartadas=array(21,22,23,25,445,554,40,53,110,135);
28.
29.  //Quantidade de caracteres disponíveis
30.  $qtdchr=strlen($chr);
31.
32.  // As variáveis sufixo e prefixo serão acrescentadas ao domínio
33.
34.  //Sufixo
35.  $sufixo="";
36.
37.  //Prefixo
38.  $prefixo="";
```

```
39.
40.  //String atual (Esse é o subdomínio)
41.  $curstr="k";
42.
43.  /* A posição atual na string $curstr. Neste caso a última
44.   * letra é p e a posição dela no alfabeto ($chr) é 15
45.   */
46.  $chrposatual=strpos($chr,substr($curstr,-1,1));
47.  $curpos=strlen($curstr)-1;//Posicao atual
48.
49.  // Inicia o contador em 0
50.  $count=0;
51.    $protocolos=array('http','https');
52.
53.
54.  // Quando true o script é finalizado
55.  $done=false;
56.  function is_html($s){
57.    //Verifica se uma string é HTML
58.    return preg_match("/<[^<]+>/",$s,$m) != 0;
59.  }
60.  do{
61.
62.    if($count>0){
63.        // Colocando o ponteiro na primeira posição do vetor chr
64.        $chrposatual=0;
65.    }
66.
67.    // Executando o for para todos os caracteres no vetor chr
68.    for($i=$chrposatual;$i<$qtdchr;$i++){
69.
70.        // Setando o caracter atual na posição atual.
71.        $curstr[$curpos]=$chr[$i];
72.
73.        /*Subdomínio formado concatenado com o domínio principal.
74.         *Ex: unip.tecvozddns.com.br
75.         */
76.        $dominio=$prefixo.$curstr.$sufixo.'.'.$dominioAlvo;
77.        $portaAtual=0;
78.
79.        echo "\nDominio: $dominio\n";
80.
```

```
81.
82.         // Verifica se existe um IP atribuído ao domínio (FQDN)
83.         $ip=gethostbyname($dominio);
84.
85.         /*
86.          * Evitando que o script faça buscas locais. E se ele retornar o
87.          * próprio domínio quer dizer que não existe IP associado.
88.          */
89.
90.         if($ip!=$dominio && $ip!='0.0.0.0' && $ip!='127.0.0.1'){
91.
92.             echo "IP: $ip\n";
93.
94.             // Salva o domínio encontrado no arquivo domains.txt
95.             file_put_contents("domains.txt","$dominio:$ip\n",FILE_APPEND);
96.
97.
98.             if(isset($nmap)){
99.                 // Apenas matando o vetor $nmap
100.                unset($nmap);
101.            }
102.
103.            /*
104.             * Executando o comando interno nmap no IP especificado e
105.             * guardando o resultado na variável $nmap
106.             */
107.
108.            exec("nmap -Pn $ip",$nmap);
109.
110.            $linhas=null;
111.            $linhas=$nmap;
112.
113.            foreach($linhas as $l){
114.
115.                /* Verifica se a linha retornada contém a palavra open,
116.                 * simbolizando uma porta aberta. Se não existir a
117.                 * palavra "open" pule para a próxima iteração
118.                 */
119.                if(strpos(trim($l),"open")<=0){
120.                    continue;
121.                }
122.
```

```
123.            if(isset($matches)){
124.                // Matando o vetor
125.                unset($matches);
126.            }
127.
128.            /* Executa a verificação por expressão regular para
129.             * capturar o número de porta no início das resposta
130.             * do nmap
131.             */
132.            preg_match("/^[0-9]{2,5}/i",trim($l),$matches);
133.
134.            // Encontrou uma porta remove os espaços do início e final
135.            $porta=isset($matches[0]) ?trim($matches[0]):false;
136.
137.
138.            /* Se a porta encontrada estiver no vetor de portas
139.             * descartadas pule para a próxima iteração
140.             */
141.
142.            if(in_array($porta,$portasDescartadas)){
143.                echo "\nEscapando porta $porta\n";
144.
145.                continue;
146.            }
147.
148.
149.            if($porta){
150.                $encontradoProto=false;
151.                $count+=1;
152.
153.                echo "\nPorta:{$porta}\n";
154.
155.                /* Testando os protocolos especificados (HTTP e HTTPS).
156.                 * Verificando se existe resposta.
157.                 */
158.                foreach($protocolos as $protocolo){
159.
160.                    /* Se já foi encontrado o serviço para o
161.                     * protocolo continue para a próxima porta
162.                     */
163.
164.                    if($encontradoProto){
```

```
165.                     continue;
166.                 }
167.
168.                 $ch = curl_init();
169.                 curl_setopt($ch, CURLOPT_URL, "{$protocolo}://{$dominio}:{$porta}");
170.                 curl_setopt($ch, CURLOPT_HEADER, 0);
171.                 curl_setopt($ch, CURLOPT_FAILONERROR, TRUE);
172.                 curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
173.                 curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 3);
174.                 curl_setopt($ch, CURLOPT_TIMEOUT, 5);
175.                 $f=curl_exec($ch);
176.                 curl_close($ch);
177.
178.                 $agora=date('Y-m-d H:i:s');
179.
180.                 // Verifica se o retorno do cURL é um texto HTML
181.                 if(is_html($f)){
182.
183.                     $encontradoProto=true;
184.                     echo "\nEncontrei usando o protocolo {$protocolo}\n";
185.
186.                     /* Registrando somente portas http bem como
187.                      * a data e hora atuais
188.                      */
189.
190.                     file_put_contents("discovered-http.txt",
191.                     "{$agora}|{$ip}|{$protocolo}://{$dominio}:{$porta}\n",
192.                     FILE_APPEND);
193.
194.                 }else{
195.
196.                     if($portaAtual!=$porta){
197.                         $portaAtual=$porta;
198.
199.                         echo "\nNada na porta {$porta} usando {$protocolo}\n";
200.                         ;
201.                         /* Registra os serviços não http encontrados
202.                          * para experiências futuras
203.                          */
204.
205.                         file_put_contents("discovered-nohttp.txt",
206.                         "{$agora}|{$ip}|{$dominio}|{$porta}\n",
```

```
207.                             FILE_APPEND);
208.                         }
209.
210.                     }
211.
212.
213.                 }
214.             }
215.
216.         }
217.
218.     }else{
219.         echo "Nenhum IP associado\n";
220.     }
221.
222. }
223.
224. /* Se chegou no fim da lista de caracteres, volta para a
225.  * primeira posicao
226.  */
227. if($i>=$qtdchr-1){
228.
229.
230.     $curstr[$curpos]=$chr[0];
231.
232.
233.     /* Se já passou da primeira posição, rodar um for para
234.      * incrementar as anteriores. Ex: az > ba > bb > bc
235.      */
236.     if($curpos>0){
237.
238.         for($j=$curpos-1;$j>=0;$j--){
239.
240.             // Descobre a posição de $j na string atual
241.             $posicao=strpos($chr,$curstr[$j]);
242.
243.             // Se a posição for igual à posição do último caractere (z)
244.             if($posicao==strlen($chr)-1){
245.
246.                 // Volta a posição para o primeiro caractere
247.                 $curstr[$j]=$chr[0];
248.
```

```
249.                }else{
250.
251.                    /* Caso contrário vá para a próxima posição. Ex:
252.                     * aaa > aab > aac
253.                     */
254.                    $curstr[$j]=$chr[$posicao+1];
255.                    break;
256.                }
257.
258.                if($j==0){
259.
260.                    $curpos+=1;
261.                }
262.
263.            }
264.        }
265.
266.        if($curpos==0){
267.
268.            //Se estiver na posição 0, vá para a próxima
269.            $curpos+=1;
270.
271.        }
272.    }
273.
274.    if($curpos>=$maxstrlen){
275.
276.        //Ponto de parada. Se atingir o máximo de caracteres
277.        $done=true;
278.
279.    }
280.
281.  }while(!$done);
282. ?>
```

7. APÊNDICE 2 - TESTADOR DE AUTENTICAÇÃO

O código fonte do script logintest.php foi omitido por conflitar com a Lei 12.737 de 30 de Novembro de 2012.