CENTRO DE INSTRUÇÃO DE GUERRA ELETRÔNICA

Cap QCO INFO JOSÉ FRANCISCO NONATO FILHO

METODOLOGIA PARA TRANSFORMAR DISPOSITIVO *UNVERSAL SERIAL BUS* (USB) CONVENCIONAL EM DISPOSITIVO *RUBBER DUCKY* USB

Brasília
2017

Cap QCO Info JOSÉ FRANCISCO NONATO FILHO

METODOLOGIA PARA TRANSFORMAR DISPOSITIVO *UNIVERSAL SERIAL BUS* (USB) CONVENCIONAL EM DISPOSITIVO *RUBBER DUCKY* USB

Trabalho de Conclusão do Curso Básico de Guerra Eletrônica para Oficiais apresentado ao Centro de Instrução de Guerra Eletrônica como requisito para obtenção do Grau de Pós-Graduação *Lato Sensu*, nível de especialização em Guerra Cibernética.

Orientador: 1º Sgt ADÃO DOS SANTOS

Coorientador: 2º Ten OTT/BIBLIO THAÍS RIBEIRO MORAES MARQUES

Brasília
2017

Ficha Catalográfica Elaborada pela Biblioteca
do Centro de Instrução de Guerra Eletrônica (CIGE)
Bibliotecária Responsável: 2° TenThaís Moraes CRB1/1922

N812m
Nonato Filho, José Francisco.
Metodologia para transformar dispositivo *universal serial bus* (usb) convencional em dispositivo *rubber ducky* usb. / José Francisco Nonato Filho – Brasília: Centro de Instrução de Guerra Eletrônica,- 2017.
38f.; il.

Trabalho de conclusão apresentado ao Curso de Guerra Cibernética para Oficiais – Centro de Instrução de Guerra Eletrônica, Brasília, 2017.

Bibliografia: f. 37.

1. Dispositivos. 2. Rubber Ducky.-USB. I Nonato Filho, José Francisco. II. Centro de Instrução de Guerra Eletrônica. III. Título.
CDD355

Cap QCO Info JOSÉ FRANCISCO NONATO FILHO

METODOLOGIA PARA TRANSFORMAR DISPOSITIVO *UNVERSAL SERIAL BUS* (USB) CONVENCIONAL EM DISPOSITIVO *RUBBER DUCKY* USB

Trabalho de Conclusão do Curso Básico de Guerra Eletrônica para Oficiais apresentado ao Centro de Instrução de Guerra Eletrônica como requisito para obtenção do Grau de Pós-Graduação *Lato Sensu*, nível de especialização em Guerra Cibernética.

Aprovado em: ___ de novembro de 2017

_____
ADÃO DOS SANTOS – 1º Sgt
Orientador

_____
THAÍS RIBEIRO MORAES MARQUES – 2º Ten
Coorientador

_____
FELIPE RODRIQUES DE VASCONCELOS - Cap
membro da comissão de avaliação

_____
VINÍCIUS EMILIANO DOS SANTOS – 2º Sgt
membro da comissão de avaliação

Brasília
2017

Dedico este trabalho primeiramente a Deus, por se essencial em minha vida, autor do meu destino, socorro presente na hora da angústia, a minha família que muito me incentivou a realizá-lo.

AGRADECIMENTOS

A Deus por ter me dado saúde e determinação para superar as dificuldades.

A minha mãe, irmãos, minha filha Giovana, meu filho José Benicio e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

A minha esposa Lucimeire, pessoa com quem amo partilhar a vida. Com você tenho me sentido mais vivo de verdade. Obrigado pelo carinho, a paciência e por sua capacidade de me trazer paz na correria durante a realização do curso.

Ao meu orientador, 1º Sgt ADÃO DOS SANTOS, por compreender a evolução na confecção do presente trabalho, sabendo adequar os prazos e tempos destinados ao mesmo.

A minha coorientadora, 2º Ten OTT BIBLIO THAÍS RIBEIRO MORAES MARQUES, pelas inúmeras orientações e dedicação, auxiliando-me durante a realização deste trabalho.

Disistir é a saída dos fracos, insistir é a vitória dos fortes. O grande prazer da vida é fazer o impossível.

Marcelo Gonçalves

RESUMO

Referência: NONATO FILHO, José Francisco. **Metodologia para transformar dispositivo *Universal Serial Bus* (USB) convencional em dispositivo *Rubber Ducky* USB**. 2017. folhas. Monografia (Curso de Guerra Cibernética para Oficiais)- Centro de Instrução de Guerra Eletrônica, Brasília, 2017.

Este trabalho tem como finalidade apresentar um passo-à-passo de como transformar dispositivo USB *(Universal Serial Bus)* em Rubber Ducky USB. A proposta foi a implementação de um *Rubber Ducky* em um dispositivo USB, no nosso caso um *pendrive* convencional, que pode ser encontrado de maneira muito fácil no mercado. A aplicação de *Rubber Ducky* foi instalada e testada em um *pendrive*, podendo ser qualquer aparelho eletrônico USB, tendo com base programas desenvolvidos em ferramentas para software livre, que contem um conjunto de aplicativos que realizam a exploração de vulnerabilidades de um alvo específico. Esses aplicativos são desenvolvidos em código aberto *(open source*), multiplataforma e que pode ser obtido gratuitamente em repositórios na rede mundial de computadores. O trabalho foi baseado na análise das técnicas de engenharia social para ser ter o acesso físico às instalações, onde ficam esses dispositivos alvos do ataque. Por meio desse trabalho pretendeu demonstrar a construção e utilização do *Rubber Ducker* para poder realizar ataques a dispositivos de informática (computadores de uso pessoal e servidores de redes) com o objetivo de comprometer ou acessar informação a esses dispositivos, utilizando o *Rubber Ducky* como dispositivo de ataque, focado como princípios e técnica de Engenharia Social.

Palavras-chave: Engenharia Social. *Rubber Ducky*. USB.

ABSTRACT

This is a Universal Serial Bus (USB) data storage program in USB Rubber Ducky. A proposal for the implementation of a Rubber Ducky in a USB device, in our case a conventional pendrive, which can be found very easily without a market. The Rubber Ducky application has been installed and tested on a USB stick, and can be any USB electronic device, based on programs developed in free software tools, which contains a set of applications that exploit vulnerabilities of a specific specific. These applications were developed in open source, cross-platform and can be obtained free of charge from repositories on the world wide web. The work in the database of the analysis of social engineering techniques for well-being around the world. Through the proposed work, demonstrate the construction and use of the Rubber Ducker to be able to carry out attacks on computer devices (personal computers and network servers) in order to compromise or access information about these devices, use Rubber Ducky as a device attack, technological focus and social engineering.

Keywords: Social engineering. *Rubber Ducky*. USB.

LISTA DE ILUSTRAÇÕES

LISTA DE SIGLAS

CFTV	CIRCUITO FECHADO DE TV
HID	DISPOSITIVO DE INTERFACE HUMANA
OSINT	OPEN SOURCE INTELLIGENTE
SDCC	SMALL DEVICE C COMPILER
USB	UNIVERSAL SERIAL BUS
TI	TECNOLOGIA DA INFORMAÇÃO

SUMÁRIO

1 INTRODUÇÃO

Vivemos em uma sociedade cada dia mais voltado à informação, que deve ser rápida, confiável e de fácil acesso, e onde a capacidade de produzir, processar e manter essa informação é requisito fundamental para determinar os rumos do crescimento e desenvolvimento de uma organização. Devido a isso, segundo Marciano e Marques (2006), várias formas de ameaças, tanto físicas quanto virtuais, proliferam-se dentro deste universo de conteúdos, que comprometem seriamente a segurança das pessoas e das informações, bem como das transações que envolvem o complexo sistema usuário-computador-informação.

Com o rápido desenvolvimento das redes de computadores – Internet no mundo e Intranet dentro de uma organização, logo, viu-se a oportunidade de buscar esse gama de informação trocada entre usuários em uma rede de computadores, as falhas de segurança físicas das instalações e aproveitando da ingenuidade das pessoas, que segundo Santos (2008), é campo mais fértil para roubo e furtos virtuais.

1.1 DELIMITAÇÃO DO TEMA

Apresentar uma metodologia para transformar dispositivo U*niversal Serial Bus* (USB) convencional em um dispositivo *Rubber Ducky* USB.

1.2 PROBLEMA

Para a realização de um estudo coerente e capaz de trazer contribuições úteis ao Exército Brasileiro, calcado na metodologia científica faz-se necessário a definição do problema para o qual será buscada uma das possíveis soluções. Isto posto, será apresentado, a seguir, como se chegou à definição deste problema.

Diante de várias fontes de informações disponíveis na internet e a gama de alvo em potencial para o ataque, visando a coleta de dados para fins de monitoramento e produção do conhecimento. Foi utilizado nesse trabalho os conceitos da engenharia social para realização de ataque em uma rede computadores críticas, com intenção de obter a informação contidas nesses equipamentos. Proponha-se neste trabalho apresentar uma metodologia (passo-a-

passo) para transformar um dispositivo USB convencional (*pendrive*) em um USB *Rubber Ducky,* como alternativa de fazer um dispositivo *Rubber Ducky* USB de custo baixo.

1.3 JUSTIFICATIVA

Diante da necessidade de transformar um dispositivo USB em *Rubber Ducky*, que permitirá obter acesso a computadores, foi proposto neste trabalho um estudo de caso que apresentou todos os passos para transformar um dispositivo USB (pendrive) em *Rubber Ducky* USB.

Com o *Rubber Ducky* USB será possível um atacante, depois do prévio conhecimento da suposta vítima, após o emprego da técnica de engenharia social, a qual foi levantado todas as informações necessárias com objetivo de ganhar acesso uma dada instalação, por exemplo *datacenter*, com a finalidade de obter dados importantes de uma organização ou indivíduo.

1.4 OBJETIVOS

Doravante serão apresentados os objetivos gerais e específicos deste estudo, estabelecendo a forma como será trabalhada a questão da aplicação da técnica da engenharia social através de produção de conhecimento em infra-estruturas críticas e com a utilizar o *Rubber Ducky* USB para ganhar o acesso a dispositivo informatizado que armazena dados importantes.

1.4.1 Objetivo Geral

O objetivo deste trabalho é apresentar uma metodologia de transformar um *pen drive* em *Rubber Ducky* USB, com a finalidade de ganhar o acesso ao dispositivo de informática que poder ser computador ou dispositivo móvel, com a finalidade de produzir informação sobre uma determinada infra-estrutura crítica de uma organização.

1.4.2 Objetivos Específicos

A fim de atingir o objetivo geral, os seguintes objetivos específicos serão buscados:

a) Abordar aspectos relevantes para a segurança da informação em uma organização;

b) Desenvolver conceito de engenharia social baseado em reflexão no tocante e definições de pesquisadores da área;

c) Abordar aspectos relevantes em demonstrar o passo-a-passo de construir um *Rubber Ducky* alternativo.

1.5 MÉTODO DE PESQUISA

O método deste estudo será a partir de pesquisa bibliográfic*a,* que de acordo com Lakatos e Marconi (1991), procura explicar o problema por intermédio de consulta a livros, trabalhos científicos, acessos sítios na internet. Ainda, de acordo com este autor, a pesquisa bibliográfica busca conhecer e analisar as contribuições culturais ou científicas existentes sobre um determinado assunto, tema ou problema.

Assim sendo, este trabalho baseou-se no estudo de caso que apresentou a elaboração do *Hubber Ducky a* partir de *pen drive*, no qual e demonstrado todos os passos de construção.

1.6 ESTRUTURA DO TRABALHO

Para facilitar o entendimento e acompanhamento do texto o trabalho está estruturado da seguinte forma: No capítulo 1 é apresentada a introdução com uma visão geral do trabalho, problema de pesquisa, objetivo geral e objetivos específicos, justificativa da importância do trabalho. No capítulo 2 é apresentada a fundamentação teórica, conceituando Segurança da Informação, Engenharia Social, Open Sources e a importância para Exército Brasileiro em demonstrar uma metodologia para construir *RUBBER DUCKY* de baixo custo. Por fim, são apresentadas as referências utilizadas no projeto de pesquisa.

2 REVISÃO DA LITERATURA

Este capítulo apresenta o referencial teórico, com a discussão de pontos de vista de autores consultados, com o objetivo de identificar posturas e idéias, por meio de uma análise crítica e reflexiva dos seus conteúdos, como aconselham Prodanov e Freitas (2013).

2.1 SEGURANÇA DA INFORMAÇÃO

A segurança da informação diz respeito à proteção de determinados dados, com a intenção de preservar seus respectivos valores para uma organização (empresa) ou um indivíduo. Podemos entender como informação todo o conteúdo ou dado valioso para um indivíduo/organização, que consiste em qualquer conteúdo com capacidade de armazenamento ou transferência, que serve a determinado propósito e que é de utilidade do ser humano.

2.1.1 Segurança da Informação em Recursos Humanos

O fator humano constitui aspecto crítico na segurança da informação, tendo em vista seu papel central em relação a processos e tecnologias. Segundo Mann (2008), a segurança humana é a conexão que falta entre segurança de TI e segurança física. Enquanto a segurança de TI está voltada para firewalls, detecção, de intrusão e antivírus, entre outros aspectos, a segurança física se volta para proteções de portas e janelas, monitoramento de áreas por meio de sistemas de Circuito Fechado de TV (CFTV) e detecção de intrusão, entre outras medidas. Para proteger seus ativos de informação, a maioria das organizações se concentra quase que completamente em segurança técnica. Atacantes detêm essa informação e muitas vezes tomam o caminho mais fácil a fim de obter informações confidenciais de uma organização: as pessoas (MANN, 2008).

2.1.2 Segurança da Informação Relacionado no Manuseio dos Equipamentos de Informáticas

Procedimentos inseguros de funcionários podem expor ativos de informação sensíveis a ações de individuo mal intencionado, vazamento não intencional e ataques de engenharia social, entre outras ameaças ao negócio da organização. À vista disso, a política de segurança da informação deve abranger responsabilidades dos usuários em relação ao manuseio de ativos de informação organizacionais.

A Cartilha de Segurança para Internet, versão 4/Cert.br abrange na seção Controle de Acessos a categoria Responsabilidades dos Usuários, a qual estabelece controles para uso de senhas, equipamento de usuário sem monitoração e política de mesa e tela limpa. O objetivo da categoria é prevenir o acesso não autorizado dos usuários e evitar o comprometimento ou roubo da informação e dos recursos de processamento da informação. Conforme dispõem os controles dessa categoria, a cooperação de usuários autorizados é essencial para uma efetiva segurança; os usuários devem estar conscientes de suas responsabilidades para manter efetivo controle de acesso, particularmente em relação ao uso de senhas e de segurança dos equipamentos de usuários; e ainda deve ser implementada política de mesa e tela limpa para reduzir o risco de acessos não autorizados ou danos a documentos/papéis, mídias e recursos de processamento da informação.

Em relação ao uso de senhas, a Cartilha preconiza que os usuários sejam solicitados a seguir as boas práticas de segurança da informação na seleção e uso de senhas. Para tal, os usuários devem manter a confidencialidade das senhas; alterar a senha se houver indícios de comprometimento do sistema ou da própria senha; utilizar senhas com caracteres alfanuméricos e especiais; modificar senhas regularmente, de forma a evitar a reutilização de senhas antigas; modificar senhas temporárias no primeiro acesso ao sistema; evitar a inclusão de senhas em processos automatizados para acessar os sistemas; não compartilhar senhas de usuários individuais; e não utilizar a mesma senha para uso profissional e pessoal.

2.2 ENGENHARIA SOCIAL

Atualmente define Engenharia Social como um conjunto de práticas, usadas

com o objetivo de obter informações importantes, explorando basicamente a confiança das pessoas. Dentro do contexto de um sistema de segurança da informação é uma forma de ataque não técnica, uma vez que não é necessário qualquer conhecimento de tecnologia. O foco é integralmente nas habilidades interpessoais, utilizadas para ganhar a confiança das pessoas utilizando-se de diversas técnicas para isso. Segundo Mann (2011), a prática de obter informações confidenciais por meio de manipulação de usuários, ou seja, técnicas para manipular pessoas, enganando-as, para que forneçam informações ou executem uma ação.

Engenharia Social é definida como conjunto de métodos e técnicas que objetiva obter informações sigilosas e importantes usando a confiabilidade das pessoas, através de técnicas investigativas, psicológicas e de enganação. Para isso, se faz passar por outra pessoa, assumindo personalidades diferentes, vasculhando lixo e outras fontes de informações, fazendo contatos com parentes e amigos da vítima e outras técnicas de obtenção de informações sigilosas (PARODI, 2008).

Todos os sistemas informatizados, dos mais simples aos mais complexos, são desenvolvidos no intuito de serem eficientes e seguros, ou seja, com o objetivo de proverem produtividade ao mesmo tempo que protegem informações. Porém, todo sistema, por mais bem planejado que seja, sempre depende de uma intervenção humana que, se for falha, pode comprometer o mais consistente dos sistemas. Sendo assim o engenheiro social utiliza-se destas falhas humanas para ganhar o acesso aos diversos sistemas de informação. O engenheiro social no contexto da segurança da informação é arte de manipulação de pessoas para levá-las inconscientemente a executar ações que causam danos à confidencialidade, integridade e disponibilidade de recursos da organização, incluindo a informação, os sistemas de informação e os sistemas financeiros (JEREMY et al, 2015).

A seguir serão apresentadas algumas técnicas utilizadas pelos Engenheiros Sociais.

Segundo Mitnick e Simon (2003):

a) **Analise do Lixo**: Provavelmente poucas organizações têm o cuidado de verificar o que está sendo descartado da empresa e de que forma é realizado este descarte. O lixo é uma das fontes mais ricas de informações para os Engenheiros Sociais. Existem muitos relatos e matérias publicadas na Internet abordando este tipo de ataque, visto que através das informações coletadas no lixo podem conter nome de

funcionários, telefone, e-mail, senhas, contato de clientes, fornecedores, transações efetuadas, entre outros, ou seja, este é um dos primeiros passos para que se inicie um ataque direcionado à empresa;

b) **Internet e Redes Sociais:** Atualmente muitas informações podem ser coletadas através da Internet e Redes Sociais sobre o alvo. Quando um Engenheiro Social precisa conhecer melhor seu oponente, esta técnica é utilizada, iniciando um estudo no site da empresa para melhor entendimento, pesquisas na Internet e uma boa consulta nas redes sociais na qual é possível encontrar informações interessantes de funcionários da empresa, cargos, amizades, perfil pessoal, entre outros;

c) **Contato Telefônico**: Com as informações coletadas nas duas técnicas acima, o Engenheiro Social pode utilizar uma abordagem via telefone para obter acesso não autorizado, seja se passando por um funcionário da empresa, fornecedor ou terceiros. Com certeza neste ponto o Engenheiro Social já conhece o nome da secretária, nome e e-mail de algum gestor, até colaboradores envolvidos na Tecnologia da Informação. Com um simples telefonema e técnicas de Engenharia Social se passando por outra pessoa, de preferência do elo de confiança da vítima, fica mais fácil conseguir um acesso ou coletar informações necessárias da organização;

d) **Abordagem Pessoal:** Está técnica consiste do Engenheiro Social realizar uma visita na empresa alvo, podendo se passar por um fornecedor, terceiro, amigo do diretor, prestador de serviço, entre outros, no qual através do poder de persuasão e falta de treinamento dos funcionários, consegue sem muita dificuldade convencer um segurança, secretária, recepcionista a liberar acesso ao *datacenter* onde possivelmente conseguirá as informações que procura. Apesar desta abordagem ser arriscada, muitos *Crackers* já utilizaram e a utilizam até hoje;

e) ***Phishing:*** Sem dúvidas esta é a técnica mais utilizada para conseguir um acesso na rede alvo. O *Phishing p*ode ser traduzido como “pescaria” ou “e-mail falso”, que são e-mails manipulados e enviados a

organizações e pessoas com o intuito de aguçar algum sentimento que faça com que o usuário aceite o e-mail e realize as operações solicitadas. Os casos mais comuns de *Phishing* são e-mails recebidos de supostos bancos, nos quais afirma que sua conta está irregular, seu cartão ultrapassou o limite, ou que existe um novo software de segurança do banco que precisa ser instalado senão irá bloquear o acesso. Outro exemplo de *phishing* pode ser da Receita Federal informando que seu CPF está irregular ou que o Imposto de Renda apresentou erros e para regularizar consta um *link*, até as situações mais absurdas que muitas pessoas ainda caem por falta de conhecimento, tais como, e-mail informando que você está sendo traído(a) e para ver as fotos consta um *link* ou anexo, ou que as fotos do churrasco já estão disponíveis no *link*, entre outros. A maioria dos *Phishings* possuem algum anexo ou *links* dentro do *e-mail* que direcionam para a situação que o Cracker deseja;

f) **Falhas Humanas**: O Ser Humano possui várias vulnerabilidades que são exploradas pelos Engenheiros Sociais, tais como, confiança, medo, curiosidade, instinto de querer ajudar, culpa, ingenuidade, entre outros.

2.3 COLETA DE INFORMAÇÕES EM FONTES ABERTAS

Nesta fase é suma importância para iniciar a coleta de todas as informações relevantes do determinado sistema-alvo. Nessa fase tem com objetivo coletar o máximo de conhecimento sobre o sistema-alvo, buscando o máximo de informações sobre ele sem atacá-los. O objetivo é reconhecer e fazer um planejamento detalhado das ameaças e de acordo com essas informações descobertas será verificada as vulnerabilidades dos sistemas-alvo.

Pode se aprender bastante sobre um indivíduo ou organização sem utilizar os dados de inteligência de fontes secretas, por exemplo, usar engenharia social, e vasculhar lixos. Tais informações são coletadas de fontes legais disponíveis na internet, por exemplo, registros públicos, publicações em jornais e procura ou oferta de emprego. O sucesso de uma invasão depende muito dos dados coletados em *Open Source Intelligence*, ou inteligência de Fontes abertas (OSINT). OSINT trata-se

do acesso a documentos oficias não classificado como de acesso restrito, informações de estado, estatísticas, livros, bem como a observação pura e simples em área de acesso permitido (CEPIK, 2003).

As fontes abertas, apesar de não englobarem documentos restritos com informações capitais, oferecem o escopo necessário para compor o compêndio inicial de dados acerca de um sistema-alvo. A vantagem da consulta desta maneira é a legalidade e a passividade em relação ao sistema – ou individuo – em foco, sendo, portanto, extremamente racional e viável.

A importância destas fontes de obtenção de dados se faz visível a partir do momento em que se vê de maneira segregada os conceitos de inteligência e segredo (AFONSO, 2006). Nem sempre o conhecimento sigiloso é reconhecido como tal por seu portador, isso somado ao fato da *internet* ter aflorado a disponibilidade, quesito indispensável quando fala em fonte aberta criou a mentalidade de se tornar acessível, quase que obrigatoriamente, aquilo que não considerado restrito.

Outra visão importante sobre fontes abertas é o conceito de que o dado não precisa ser obtido de forma clara e inteira de uma vez, ele pode estar contido nas entrelinhas. Portanto, a busca pode ser qualificada pelo sigilo de sua fonte, o que faz diferença em se tratado de informações inteiramente acessíveis é a pesquisa eficiente e metódica, com o correto emprego das ferramentas disponíveis.

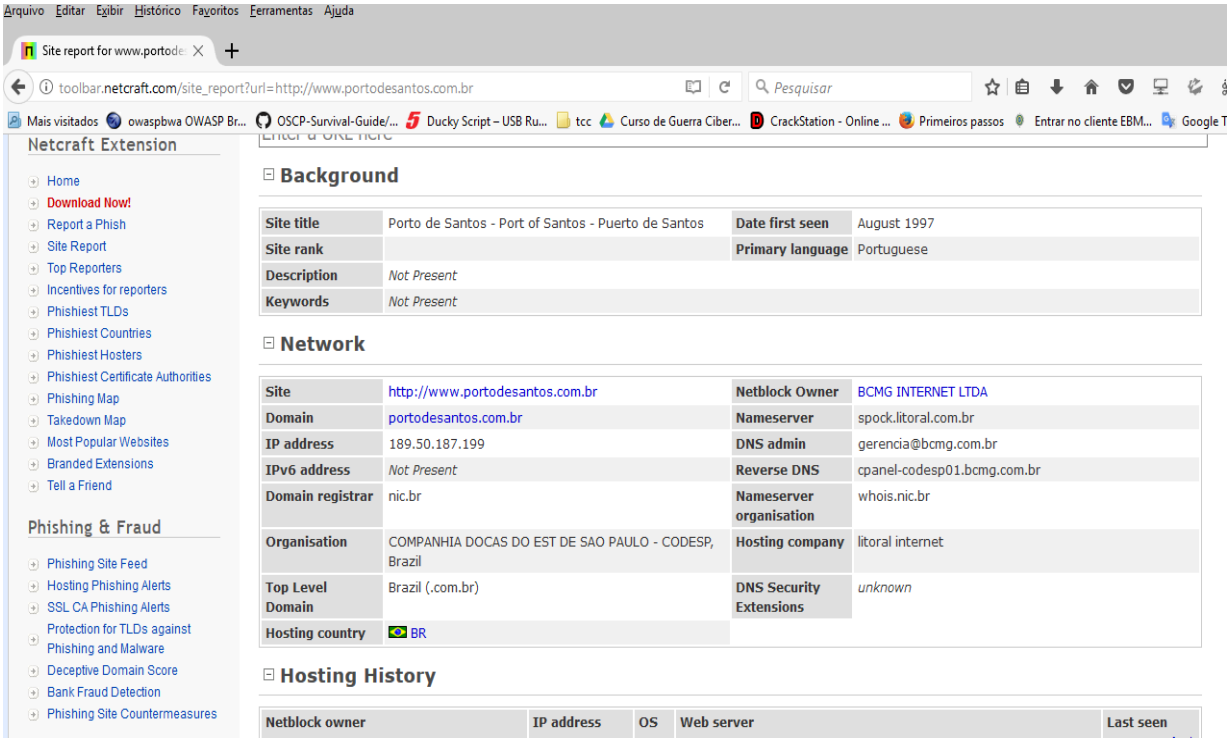
Será apresentada, a seguir, como fonte de conhecimento algumas ferramentas, entre várias, onde se poderão obter informações interessantes, oriundas dessas fontes abertas.

2.3.1 Netcraft

As informações que os servidores web e as empresas de hospedagem Web estão publicamente disponíveis e podem dizer muito a respeito de um site. Então uma empresa inglesa chamada Netcraft faz o registro do *uptime* (tempo em atividade) faz consultas sobre o software de um domínio da internet. Para realizar a consulta sobre um domínio basta acessar o site disponível em “http://www.netcraf.com/”. O resultado da consulta retorna muitas informações sobre um domínio, por exemplo, o sistema operacional do servidor que hospeda este domínio.

Segundo Weidman (2014), o Netcraft também provê outros serviços, e suas ofertas estão relacionadas ao *antiphishing* que são de grande interesse para a segurança da informação. Observe a imagem da figura 1 em que faz a demonstração de uma pesquisa de um domínio com Netcraft:

Figura 1 – Captura de tela de uma consulta do Netcraft



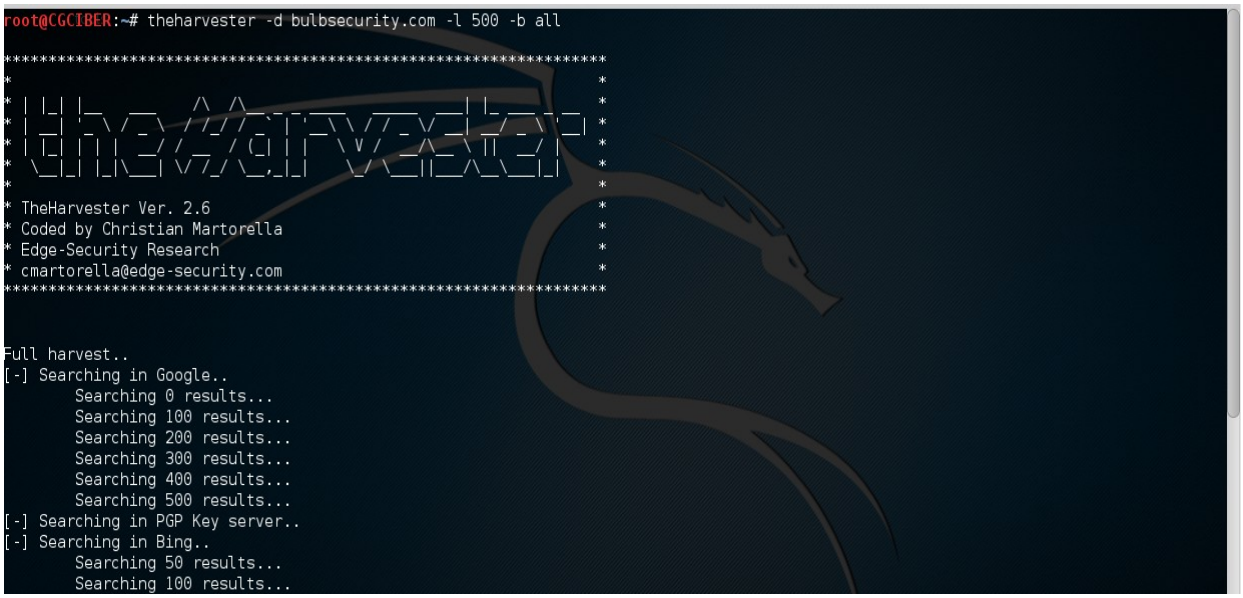
FONTE: Próprio autor (2017)

2.3.2 Theharvester

É uma ferramenta que fornece informações sobre contas de e-mail, nomes de usuários e hostnames/domínios à partir de diversas fontes públicas, como motores de buscas (Google, Bing) e servidores de chaves PGP. O *TheHarvester* estão disponíveis na maioria das distribuições Linux. Pode-se visualizar na figura 3 a execução de uma consulta com *TheHarvester*:

> Procurar endereços de email na Internet é uma maneira excelente de descobrir nomes de usuários. Você ficaria surpreso ao encontrar endereços corporativos de email listados publicamente (WEIDMAN, 2014).

Figura 2 – Captura de tela de uma pesquisa utilizando Theharvester

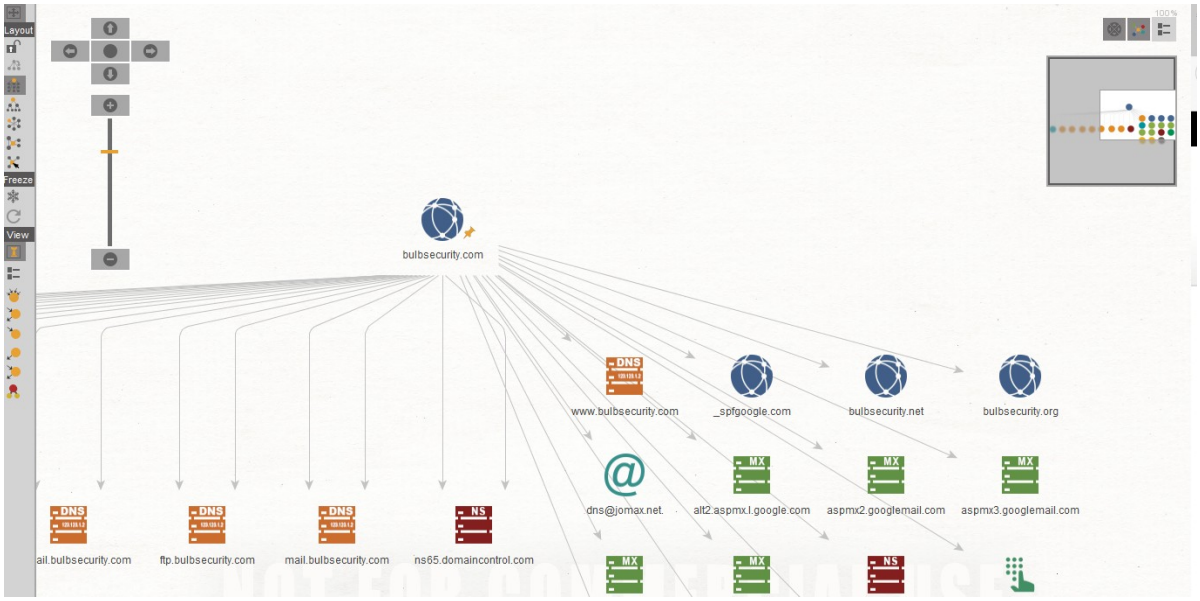


FONTE: Próprio autor (2017)

2.3.3 Maltego

O Maltego é uma ferramenta que recolhe informações de várias fontes públicas, foi projetado para visualizar o resultado da coleta de dados de inteligência de fontes abertas. O Maltego possui uma versão comercial paga que oferece mais resultados e mais funcionalidades e um versão gratuita que está disponível no Kali Linux, que o resultado retornado é limitado. Informações sobre o Maltego pode ser encontrado no site da Paterva. Observe a imagem da figura 3 em que faz a demonstração de uma pesquisa de um domínio com Maltego:

Figura 3 – Captura de tela de uma pesquisa de OSINT com Maltego



FONTE: Próprio autor (2017)

2.4 ANALISANDO AS VULNERABILIDADES E EXPLORANDO AS FALHAS

De acordo com o conhecimento obtido na fase de coleta de informações de um sistema-alvo, o atacante começa a descobrir ativamente as vulnerabilidades a fim determinar suas estratégias de exploração das falhas para ser bem-sucedidas. Na análise de vulnerabilidades o atacante executa várias ferramentas ou utiliza de técnicas que ajudará o atacante na escolha da melhorar ferramenta ou técnica manuais para identificação de vulnerabilidades.

Depois da análise o atacante faz a exploração das falhas, executando exploits contra as vulnerabilidades descobertas. Como será apresentado nos capítulos seguinte ferramentas, ou melhor, dizer, técnica de explorarão de vulnerabilidades que apresentarão algumas falhas de sistemas ou algumas falhas humanas que é o elo fraco para se obter acesso a um sistema-alvo.

Quando se ganha um sistema-alvo, o atacante reúnem as melhores informações sobre este sistema invadido, onde o mesmo poderá manter o acesso no sistema fazendo a elevação do nível de privilégios para manter sempre seu acesso no sistema.

Segundo Weidman (2014) um *exploit* bem-sucedido pode conduzir a uma fase de pós-exploração das falhas, de modo a se descobrir informações adicionais, obtendo dados críticos e acessando outros sistemas e assim por diante.

BADUSB (*RUBBER DUCKY*)

A finalidade deste trabalho é apresentar ao leitor a construção, ou melhor, a transformação de dispositivo USB ou unidade *flash* USB em dispositivo USB malicioso conhecido com BADUSB *(RUBBER DUCKY*) para ser utilizado em um sistema-alvo.

Existem várias ferramentas de invasão que pode ser utilizadas para se ganhar um sistema-alvo. A escolha da utilização do BADUSB torna-se necessário quando as outras ferramentas de exploração falham na exploração deste sistema-alvo. Este *exploit* é considerado como umas das piores ameaças à segurança já vistas. O BADUSB é praticamente indetectável e bem difícil de ser removido, pois o grande detalhe é que alguns softwares antivírus são praticamente inúteis contra ele.

O BADUSB não infecta apenas computadores, ele infecta praticamente qualquer dispositivo USB conectado a ele.

Neste capítulo será apresentado como transformar um *pendrive* em um dispositivo BADUSB e como este dispositivo malicioso poderá ser aplicado na prática.

3.1 COMO TRABALHA UM *FLASH DRIVE* (*PENDRIVE*)

Na verdade, o sistema operacional não sabe nada sobre o dispositivo conectado. Tem que esperar até que o dispositivo informe ao sistema operacional qual é o tipo de dispositivo. Considere um exemplo simples, quando se conecta uma unidade *flash* USB a uma porta USB, a unidade *flash* informa ao sistema operacional o tipo e o tamanho do volume.

O propósito dos dispositivos USB é definido pelos códigos de classe comunicados ao *host* para instalação dos *drivers* necessários. Os códigos de classe permitem que o *host* funcione com dispositivos de diferentes fabricantes. O dispositivo pode suportar uma ou várias classes, desmonstrada na fugura 4, cujo número é determinado pelo número de pontos da extremidade USB. Quando conectado, o *host* solicita uma variedade de detalhes padrão dos dispositivos (chamado de descritores), que ele usa para decidir como trabalhar com ele. Os descritores contêm informações sobre o fabricante e o tipo de dispositivo, que o *host* usa para selecionar o *driver* do programa.

Uma unidade flash USB normal terá código de classe 08h (Dispositivo de armazenamento em massa - MSD), enquanto uma câmera *web* equipada com um microfone terá dois códigos: 01h (Áudio) e 0Eh (classe de dispositivo de vídeo).

Figura 4 – Device classes

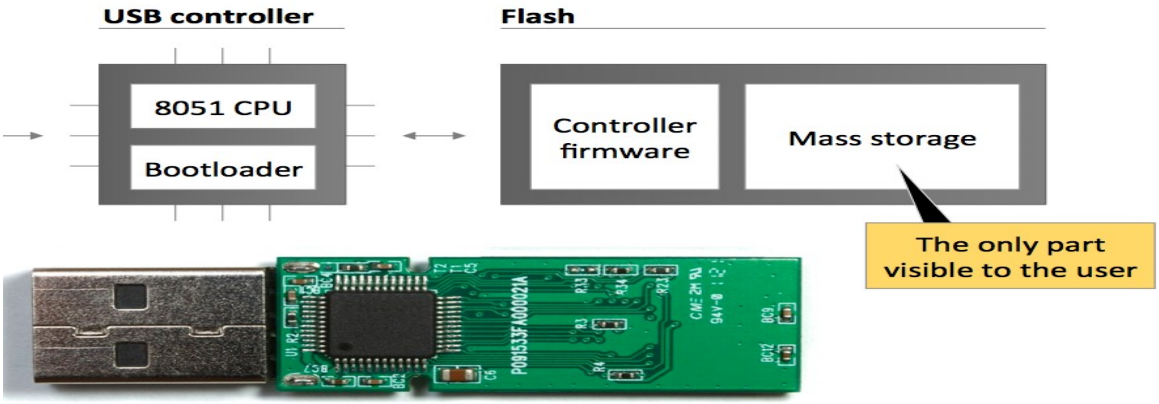
	Examples	
Identifier	USB thumb drive	Webcam
Interface class	8 – Mass Storage	a. 1 – Audio b. 14 – Video
End points	0 – Control 1 – Data transfers	0 – Control 1 – Video transfers 6 – Audio transfers 7 – Video interrupts
Serial number (optional)	AA627090820000000702	0258A350

FONTE: Anton; Zhukov (2017)

Quando conectado, o dispositivo USB é registrado, recebe um endereço e envia seu descritor para permitir que o sistema operacional instale os *drivers* (programas) necessários. Depois disso, o *host* imediatamente começa a trabalhar com o dispositivo. Uma vez concluído o trabalho, o dispositivo é desregistado. É importante notar que os dispositivos podem ter vários descritores, eles também podem se registrar e se desregistrar como um dispositivo diferente.

Se você abrir o corpo de uma unidade *flash* USB, além do armazenamento em massa visível para o usuário, há um controlador responsável pelas ações acima descritas. Pode-se observar o detalhe de uma unidade *flash* USB ilustrado na figura 5:

Figura 5 – Detalhe de uma unidade fash USB



FONTE: Anton; Zhukov (2017)

3.2 FUNCIONAMENTO DE UM *BADUSB (RUBBER DUCKER)*

Vários especialistas em segurança da informação têm demonstrado que os dispositivos USB pode ser usados para infectar um computador como malware sem ser detectado e este malware pode interceptar a internet de um usuário e instalar um software mal intencionado no computador. Isto ocorre por que o problema não está na memória do dispositivo USB ou no computador e sim no *firmware* que controla a função básica do dispositivo USB. Por exemplo, um dispositivo USB nesse caso um pendrive contem um pequeno computador e um *firmware* que é essencialmente um software especial ou um pequeno sistema operacional que faz que o dispositivo funcionar normalmente como se fosse um dispositivo USB de armazenamento de dados.

Este *firmware* funciona em baixo nível e usualmente programado na fábrica e no momento em que se adquire um *pendrive* se confia que este *firmware* funcione como foi desenvolvido pelo fabricante.

Em muitos dispositivos USB os *firmwares* pode ser reprogramados. Isto é possível porque alguns dos fabricantes não desenvolveram a segurança necessária que evite a reprogramação do *firmware*.

Como conseqüência estes dispositivos USB quando alterados pode ser usado para transmitir *malware*, como cavalos de tróia ou vírus, emular um teclado normal, enviar imagem de um *webcam* a outro computador.

3.3 PROCESSO DE TRANSFORMAÇÃO DE DISPOSITIVO USB EM UM BADUSB

A maioria dos USB comuns são explorável devido uma vulnerabilidade BADUSB. Isto permite a um *hacker* reprogramar o micro-controlador destes para que atuem como dispositivos de interface humana (HID). Neste estudo será demonstrado o micro-controlador de uma unidade USB, compilar o código fonte por um ferramenta que está publicado no Github, fazer um *firmware* customizado com um *payload* de HID incorporada e fazer a converteçaão de uma unidade USB inofensiva em um teclado malicioso desenvolvido para ajudar a atacar um equipamento de uma vítima. O processo é semelhante ao processo de compilar a *ROMs* de um dispositivo Android.

Material empregado no estudo:

a) Computador com Windows 7 ou superior;
b) Uma unidade USB 3.0 com Phison 2303(2251-03) de micro-controlador. O Toshiba TransMemory-MX USB 3.0 de 8Gb.
c) Pode se abrir a caixa de seu dispositivo USB sem danificá-lo.

3.3.1 Determinção da compatibilidade do *Firmware* do USB

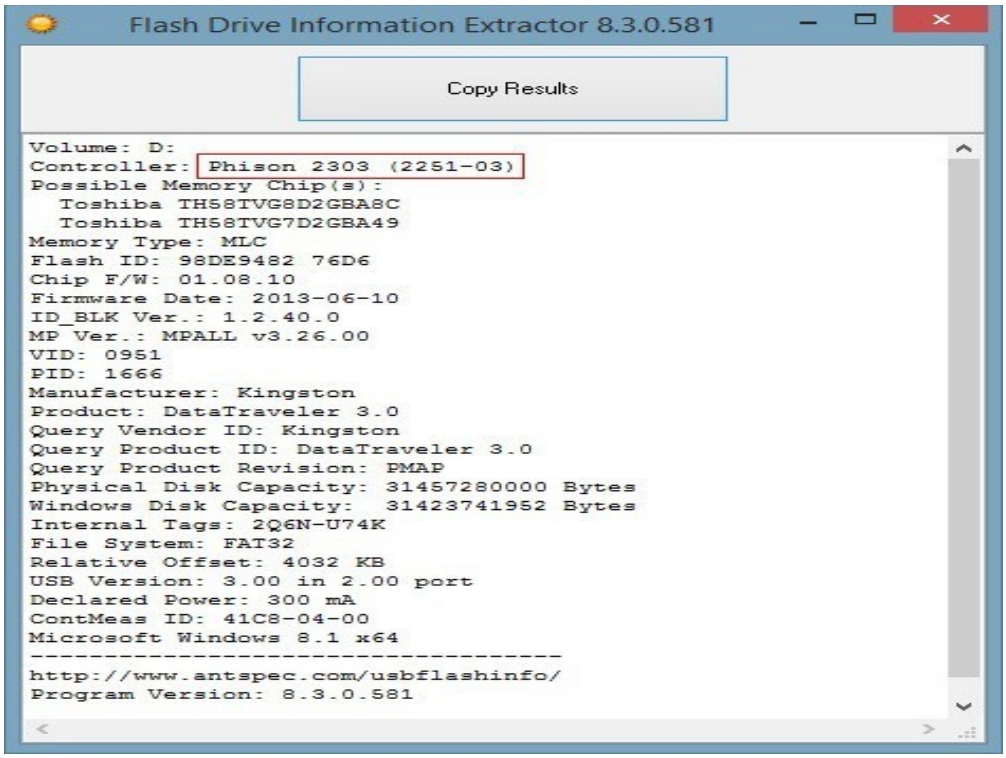
Antes de começar deve-se assegurar que o USB utiliza o controlador compatível (*firmware*). Pode-se usar um programa chamado *Flash Drive Information Extractor* para mostrar a informação de configuração do dispositivo USB.

Pode fazer o download do programa através do link abaixo:

http://www.antspec.com/files/usbfashinfo.zip.

Não precisa instalar o programa, só é preciso descompactar e abrir a ferramenta e clicar no botão “Obter Informação *USB Flash Drive*”, no entanto o dispositivo USB deverá está inserida em uma porta USB do computador. Pode-se observar na figura 6 o resultado da informação do dispositivo USB:

Figura 6 – Captura de tela obter informação de dispositivo USB



FONTE: Próprio autor (2017)

3.3.2 Configuração do ambiente de trabalho

O desenvolvedor do *exploit* recomenda que se deve usar uma das versões do "Microsoft Visual Studio 2012" ou superior para compilar as ferramentas.

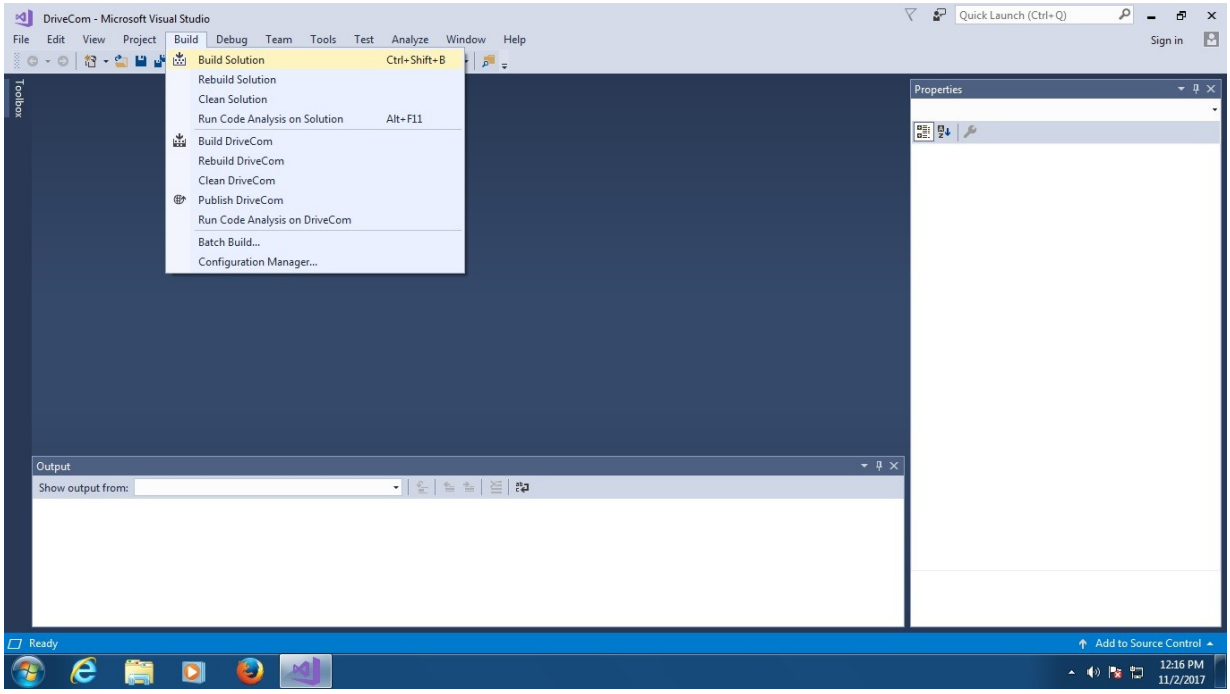
E deve fazer o download do SDCC (*SMALL DEVICE C COMPILER*) para criar o *firmware* personalizado.

3.3.3 Obtendo e compilando o código Fonte

Tendo encontrado o dispositivo adequado, pode começar a sua transformação. Primeiro de tudo, precisa baixar os códigos fontes disponibilizados no GitHub: https://github.com/adamcaudill/Psychson.

Baixe o arquivo "**.zip"** desde github, descomprimi os arquivos em um diretório de seu computador e compile os arquivos "**.sln**" dos projetos DriveCOM e EmbedPayLoad com o Visual Studio, conforme a figura 7:

Figura 7 – Captura de tela compilando projeto no Visual Studio 2012



FONTE: Próprio autor (2017)

Depois de executado os procedimentos acima, será necessário os aplicativos detalhado abaixo para transformação do dispositivo USB.

a) DriveCom – um aplicativo para comunicação com *Phison USB flash drives*;

b) EmbedPayload – um aplicativo para incorporar *Rubber Ducky* inject.bin scripts de chave em firmware personalizado para execução subseqüente quando a unidade flash USB está conectada
c) Injector - um aplicativo que extrai endereços do firmware e incorpora o código de correção no firmware;
d) Firmware – firmware 8051 personalizado escrito em C; e
e) Patch – coleção de *patches* 8051 escrito em C.

3.3.4 Obtendo *Burner Image*

Uma *Burner Image* é necessária para fazer *dumping* e atualizar o *firmware* da USB, estes são normalmente utilizados uma conversão “BNxxVyyyz.BIN”. Onde xx é o controle da versão, yyy é o número da versão e z refere-se o tamanho da pagina da memória.

A Burner Image para o firmware Phison está disponível em russo no sitio: http://www.usbdev.ru/files/phison.

3.3.5 *Dumping* do fir*m*ware original

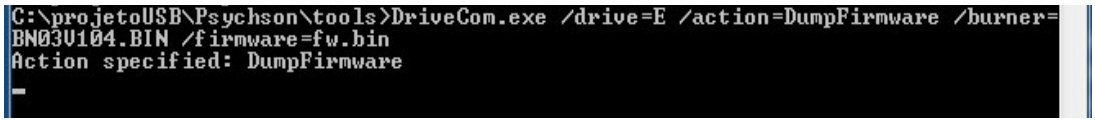
Antes de começar suas experiências é necessario fazer um backup do *firmware* original para poder recuperar caso sai algo errado com seu dispositivo. Primeiro mude seu dispositivo para modulo de inicialização, visualizado na figura 8 abaixo:

Figura 8 – Captura de tela mudança de módulo de inicialização

FONTE: Próprio autor (2017)

Em seguida, use a aplicação DriveCom, passando a letra da unidade, o caminho para a Burner Image e o caminho para o arquivo onde o firmware original será salvo, onde pode ser observado no comando executado conforme a figura 9:

Figura 9 – Captura de tela *Dump* de *Firmware*



FONTE :Próprio autor (2017)

3.3.6 Preparação *do payload*

Agora é hora de pensar sobre as funções que se quer dar para dispositivo *USB*. Para o *Rubber Ducky USB*, há um sitio inteiro, com uma interface amigável, que permite criar scripts para o seu dispositivo. Felizmente, os *scripts Ducky* podem ser convertidos em binários para inseri-los, então, no *firmware*. Para fazer isso, será preciso usar um utilitário *Duck Encoder* que está disponivel no sitio:

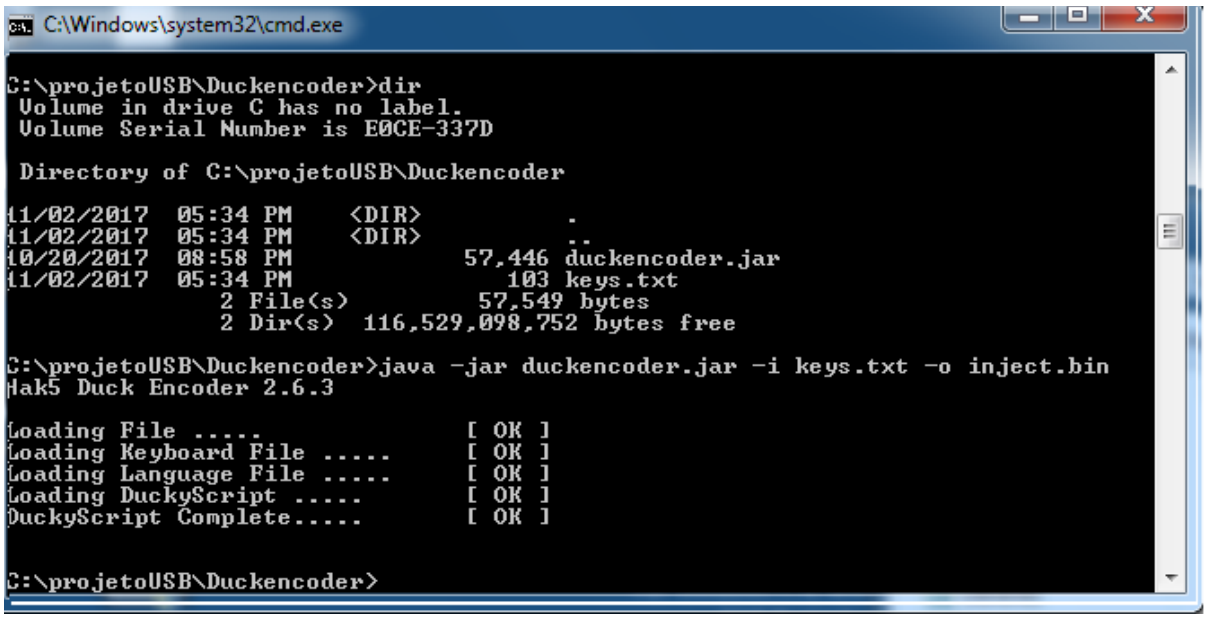
https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Downloads.

Também será necessário instalar o Java.

Quanto aos scripts, existem várias opções:

Pode-se escrever seu próprio script, pois a sintaxe usada é fácil de gerenciar (veja o site oficial do projeto). Para converter o script em binário, execute o seguinte comando, conforme figura 10:

Figura 10 – Captura de tela conversão de *script Ducker* em binário

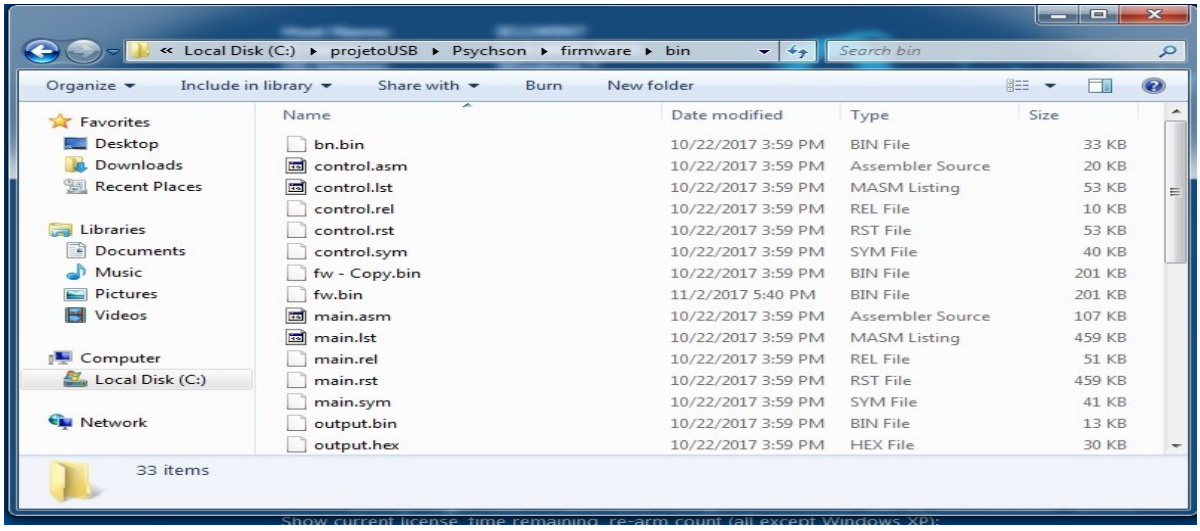


FONTE: Próprio autor (2017)

Onde Keys.txt é um *script Ducky* e inject.bin é o arquivo binário gerado.Criando um Firmware personalizado

Neste ponto todos os preparativos já estão certos. Entre no diretório Pychson/firmware e execute o arquivo build.bat. Se tudo ocorrer bem, será mostrada uma nova pasta com muitos arquivos diferentes em seu interior, conforme figura 11:

Figura 11 – Captura de tela criando *Firmware* Personalizado



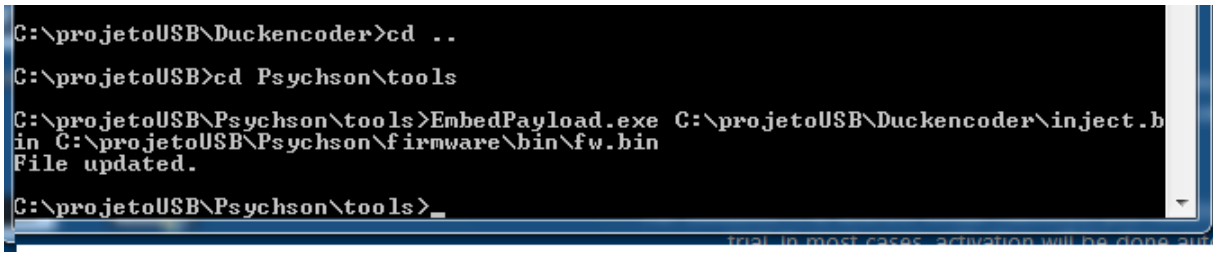
FONTE: Próprio autor (2017)

O arquivo “fw.bin” é o arquivo que será usado a seguir.

3.3.7 Inserindo o payload no *firmware*

Agora será utlizado a ferramentas construídas com Visual Studio. Obviamente *EmbedPayload* é para inserir o payload. Simplesmente será executado o comando, conforme ilustrado na figura 12:

Figura 12 – Captura de tela inserido Payload no Firmware



FONTE: Próprio autor (2017)

Finalmente tem um arquivo de firmware personalizado “fw.bin “ com um payload incorparado. Agora é somente colocar o firmware no dispositivo USB, executando o comando ilustrado na figura 13:

Figura 13 – Captura de tela transformando dispositivo USB em dispositivo HID

FONTE: Próprio autor (2017)

O dispositivo USB foi convertido em um dispositivo HID.

3.4 APLICAÇÃO PRÁTICA DO BADUSB (*RUBBER DUCKY*)

A legislação brasileira considera o ato de invadir computadores como crime passível de punição. Cada país conta com as suas normas e regras quando a assunto é crime cibernético (BORGES, 2013).

No Brasil a lei de n. 12.737 (apelidada de Carolina Dieckman):

> 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 tática do titular do dispositivo ou instalar vulnerabilidades para obter vantagens ilícita:
> Pena – detenção, de 3 (três) meses a 1(um) ano, e multa. (BRASIL, 2013 apud BORGES, 2013).

Devido a características deste trabalho vale mencionar que ataques digitais é crime no Estado Brasileiro. Portanto cabe lembrar ao leitor, invadir dispositivo de informática, incluído acesso não autorizado a redes, é considerado crime de acordo com a legislação brasileira. Deve-se considerar a legislação em vigor para poder descrever uma aplicação prática de invasão a sistemas-alvo vulneráveis de uma organização que será descrita neste capitulo.

Então, para acessar um sistema-alvo vulnerável, o atacante após um minucioso estudo das áreas vulneráveis nas instalações de uma organização e também após a avaliação dos tipos de pessoas que trabalham ou freqüentas estas instalações. Tal processo inicia-se durante as fases de conhecimentos e coletas de dados através da engenharia social. De posse das informações descrita sobre um determinado sistema-alvo, o atacante poderá acender ao prédio e instalar um BADUSB (*Rubber Ducky*) com o propósito de conectar esse dispositivo malicioso a um *host* (qualquer máquina ou computador conectado a uma rede) para poder ter acesso a algum tipo de informação dessa organização.

Uma maneira de como um atacante poderá acessar uma instalação é por intermédio de uma credencial de acesso (crachá). Este distintivo de acesso poderá trazer muita informação sobre o empregado da empresa, tais como: nome, cargo, departamento, fotografia e logotipo da empresa e alguns deles possui faixa magnética que permite que o empregado se autentique para passar por certos níveis de controles físicos. O atacante poderá utilizar-se deste distintivo (crachá) para conseguir o acesso físico a uma instalação. O atacante poderá através de técnicas

de engenharia social criar um distintivo ou também poderá furtá-lo com o propósito de ter uma credencial de acesso para as instalações da empresa em questão.

Existem várias técnicas no qual um atacante poderá utilizar para se ter acesso a uma instalação. Então neste trabalho deve-se focar na instalação de BADUSB com um código malicioso que será conectado a um computador através de uma porta USB.

Na prática a implantação do BADUSB em computador poderá ser feito de várias maneiras: uma delas é através de acesso as instalações físicas por um profissional bem treinado e espertar o BADUSB em um *host* com um *payload* injetado que abrirá um *backdoor* para tentar explorar este *host* ou rede. Outra maneira de se conseguir instalar o BADUSB é por intermédio do elemento mais vulnerável na segurança da informação, que é o fator humano. O atacante poderá escolher um alvo e estudar sua rotina após um minucioso trabalho de engenharia social e influenciar este indivíduo a conectar BADUSB com um *malware* no seu computador.

O BADUSB poderá usar por exemplo, um *exploit*, que abrir uma janela de comando e fazer o download e instalação de software malicioso na maquina no qual foi conectado, ou então transforma-se em uma interface de rede que, uma vez ativa, fará com que o computador se conecte a sítios maliciosos disfarçados por legítimos sobre o controle do atacante.

Executar um ataque físico a uma empresa e instalar um BADUSB em um computador exige estar fisicamente no local. Portanto, torna-se uma tarefa difícil e delicada e que exige do atacante fazer uso de alguns adereços visuais, disfarces e falsificação ou furto de credenciais.

4 CONCLUSÃO

Neste trabalho foi apresentado a transformação de um dispositivo de armazenamento USB em um BADUSB HID para atacar um sistema de informação de uma organização. Este dispositivo malicioso pode ser utilizado por um indivíduo treinando para esta tarefa, que por meio de acesso físico a uma instalação poderá conectar o dispositivo em computador-alvo.

Face o objetivo inicial, não foi só constituído uma prova de conceito que demonstra a aplicabilidade da solução proposta e desenvolvida, como foi criado um trabalho precursor e passível de melhoria no futuro, estabelecendo uma base daquilo que vai ser fundamental na área de ataque de cibernética. Apesar de simples, o trabalho representou um acréscimo de valor nas atividades da guerra cibernética.

Contudo, cabe lembrar que a transformação de um *pendrive* em um BADUSB é uma grande ameaça aos sistemas de informação. Este dispositivo quando empregado de forma desordenada poderá infectar vários computadores ou equipamentos que utilizam porta USB, pois o *malware* fica injetando no firmware do *pendrive*, que é praticamente indetectável e difícil de ser removido.

Para além do esforço de investigação este trabalho foi mais um passo no processo continuo de crescimento e aprendizado, aliado ao conhecimento teórico adquirido ao longo da pesquisa, tornando possível a realização do estudo da ferramenta para transformar um dispositivo de armazenamento USB em dispositivo malicioso que poderá ser utilizado em ataque cibernético a um computador ou uma rede computadores.

REFERÊNCIAS BIBLIOGRÁFICAS

ANTON, ZHUKOV. **Turning a Regular USB Flash Drive into a USB Rubber Ducky**. 2017. Disponível em: < https://hackmag.com/security/rubber-ducky/https://hackmag.com/security/rubber-ducky/>. Acesso em: 20 out. 2017.

BORGES, Abimael. **Lei Carolina Dieckmann – Lei nº 12.737, art. 154-A do Código Penal**. 2013. Disponível em: <https://abimaelborges.jusbrasil.com.br/artigos/111823710/lei-carolina-dieckmann-lei-n-12737-12-art-154-a-do-codigo-penal>. Acesso em: 31 out. 2017.

LAKATOS, E. M.; MARCONI, M. A. **Fundamentos de metodologia científica.** 5. ed. São Paulo: Atlas, 2003.

MARCIANO, J. L.; MARQUES, M. L. O Enfoque Social da Segurança da Informação**. Revista Ciência da Informação**, Brasília, v. 35, n. 3, p. 89-98, set./dez. 2006.

MITNICK, K. D.; SIMON, W. L. **A Arte de Enganar:** ataques de hackers – controlando o fator humano na segurança da Informação. São Paulo: Makron, 2003.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar. **Metodologia do Trabalho Científico**: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico. 2. ed. Novo Hamburgo, RS - Rio Grande do Sul: Feevale, 2013.

SANTOS, A. H. G. **A história da segurança da informação**. 2008. Disponível em: <http://www.slideshare.net/andrehor/a-histria-da-segurana-da-informao-presentation-902879>. Acesso em: 30 set. 2017.

WEIDMAN, Georgia. **Testes de Invasão**. Uma introdução prática ao hacking. 1. Ed. São Paulo: Novatec, 2014.

GLOSSÁRIO

BACKDOOR é um recurso utilizado por diversos malwares para garantir acesso remoto ao sistema ou à rede infectada, explorando falhas críticas não documentadas existentes em programas instalados, softwares desatualizados e do firewall para abrir portas do roteador.

DATACENTER ambiente projetado para abrigar servidores e outros componentes como sistemas de armazenamento de dados e ativos de rede.

EXPLOITS são um subconjunto de malware. Normalmente, são programas maliciosos com dados ou códigos executáveis capazes de aproveitar as vulnerabilidades de sistemas em um computador local ou remoto.

FIRMWARE é o conjunto de instruções operacionais programadas diretamente no hardware de um equipamento eletrônico.

HOST é qualquer máquina ou computador conectado a uma rede, podendo oferecer informações, recursos, serviços e aplicações aos usuários ou outros nós na rede.

JAVA linguagem de programação e plataforma computacional lançada pela primeira vez pela Sun Microsystems em 1995.

MALWARE é considerado um tipo de software irritante ou malígno que pretende acessar secretamente um dispositivo sem o conhecimento do usuário. Os tipos de malware incluem spyware, adware, phishing, vírus, Cavalos de Tróia, worms, rootkits, ramsoware e sequestradores de navegador.

PATCH programa de computador criado para atualizar ou corrigir um software.

PAYLOAD refere-se à carga de uma transmissão de dados.

PENDRIVE dispositivo de memória constituído por memória flash com aspecto semelhante a um isqueiro utilizado para armazenar dados.

SCRIPT é uma linguagem de **programação** que suporta **scripts**, programas escritos para um sistema de tempo de execução especial que automatiza a execução de tarefas que poderiam alternativamente ser executadas uma por vez por um operador humano.