DOI:

VULNERABILITY EXPLOITATION IN THE SMBV1 PROTOCOL

EXPLORAÇÃO DA VULNERABILIDADE NO PROTOCOLO SMBV1

Wellington Sousa Aguiar
CENTRO UNIVERSITÁRIO ESTÁCIO DO CEARÁ - ORCID: https://orcid.org/0000-0003-0677-5782
Alexandre Bonadiman Angeli
FACULDADE ESTÁCIO DE SÁ DE VITÓRIA - FESV - ORCID: https://orcid.org/0000-0002-7992-3195
José Mário Bezerril Fontenelle
CENTRO UNIVERSITÁRIO ESTÁCIO DO CEARÁ - ORCID: https://orcid.org/0000-0001-6828-0123
David Cavalcante Da Silva
CENTRO UNIVERSITÁRIO ESTÁCIO DO CEARÁ - ORCID: https://orcid.org/0000-0003-1345-1700

Abstract
Presenting the vulnerability in the SMBv1 protocol, identified by the fix released by Microsoft: MS17-010, using the EternalBlue/DoublePulsar exploit, this vulnerability had its propagation through WannaCry that using the flaw had its spread on May 12, 2017.
No system is 100% secure, information security rule, however in the field of intrusion testing most of the flaws exploited by CVE (Common Vulnerabilities and Exposures) have already been corrected and released correction path, but without updating by users .
The methodology used was research applied in a field study, through the analysis of this vulnerability and description of the entire process of exploiting the security flaw. Bibliographic research was also used to give theoretical support to the study.
With this study, it was possible to present real cases of the risk existing in outdated systems, which allow access and exploitation by malicious people.
The exploitation of vulnerabilities by ethical researchers is a safe way to help information security in organizations that rely on science to guarantee their computing assets.
The research presents the problem and the solution to guarantee information security in organizations that are vulnerable to the studied attack method, thus helping Information Security for IT managers.

Key words: Information security, SMB, MS17-010, Eternalblue, Doublepulsar

Resumo
Apresentar a vulnerabilidade no protocolo SMBv1, identificada pela correção lançada pela Microsoft: MS17-010, utilizando o exploit EternalBlue/DoublePulsar, essa vulnerabilidade teve sua propagação através do WannaCry que utilizando a falha teve sua disseminação em 12 de maio de 2017.
Nenhum sistema é 100% seguro, regra da segurança da informação, entretanto no campo de testes de intrusão a maioria das falhas exploradas por CVE (Common Vulnerabilities and Exposures) já foram corrigidas e lançado path de correção, mas sem a atualização por parte dos usuários.
A metodologia utilizada foi a pesquisa aplicada em um estudo de campo, através da análise dessa vulnerabilidade e descrição de todo o processo de exploração da falha de segurança. Foi utilizada também a pesquisa bibliográfica para dar sustentação teórica ao estudo.
Com esse estudo foi possível apresentar casos reais do risco existente em sistemas desatualizados, que permitem o acesso e exploração por pessoas mal-intencionadas.
A exploração de vulnerabilidades por parte de pesquisadores éticos é uma forma segura de auxiliar a segurança da informação nas organizações que se apoiam na ciência para garantir seus ativos computacionais.
A pesquisa apresenta o problema e a solução para garantir a segurança da informação em organizações que estão vulneráveis ao método de ataque estudado, auxiliando assim a Segurança da Informação para gestores de TI.

Palavras-chave: Segurança da informação, SMB, MS17-010, Eternalblue, Double pulsar

VULNERABILITY EXPLOITATION IN THE SMBv1 PROTOCOL

No system is 100% secure. This is one of information security statements, however within the field of intrusion testing, we found that most of the flaws exploited by CVE (Common Vulnerabilities and Exposures) have already been corrected and a correction path has been launched, but without being updated by part the users. The purpose of this article is to present the vulnerability in the SMBv1 protocol, identified by the fix released by Microsoft from MS17-010, using the EternalBlue / DoublePulsar exploit, this vulnerability was spread through WannaCry which, using the flaw, was spread on May 12, 2017. Due to this critical issue the fix was released by Microsoft on March 14, 2017, a month before the malware spread. On January 15, 2020, when I was writing this article, the lack of updating was again allowing the exploitation of this flaw that had been long identified. The methodology used was the research applied in a field study, through the analysis of this vulnerability and description of the entire process of exploiting the security breach. With this study it was possible to present the existing risk in outdated systems, which allows access and exploitation by malicious people. After the study, we can conclude that there is a great need for attention on the part of the analysts involved in the systems and infrastructure of an organization to be always attentive to the update paths , because failures that allow the injection of malicious codes can bring incalculable losses to a company , such as: loss of confidential data, file encryption, attack vectors, bank access and damage to the attacked organization image.

Keywords: Information security, SMB, MS17-010, Eternalblue/DoublePulsar

EXPLORAÇÃO DA VULNERABILIDADE NO PROTOCOLO SMBv1

Nenhum sistema é 100% seguro. Esta é uma das afirmações da segurança da informação, entretanto dentro do campo de testes de intrusão, verificamos que a maioria das falhas exploradas por CVE (*Common Vulnerabilities and Exposures*) já foram corrigidas e lançado *path* de correção, mas sem a atualização por parte dos usuários. O objetivo deste artigo é apresentar a vulnerabilidade no protocolo SMBv1, identificada pela correção lançada pela Microsoft de MS17-010, utilizando o *exploit* EternalBlue/DoublePulsar, essa vulnerabilidade teve sua propagação através do *WannaCry* que utilizando a falha teve sua disseminação em 12 de maio de 2017. Devido a este problema crítico a correção foi lançada pela Microsoft em 14 de março de 2017, um mês antes da propagação do *malware*. No dia 15 de janeiro de 2020, quando escrevia esse artigo, a falta de atualização permitia mais uma vez a exploração dessa falha há tanto tempo identificada. A metodologia utilizada foi a pesquisa aplicada em um estudo de campo, através da análise dessa vulnerabilidade e descrição de todo o processo de exploração da falha de segurança. Com esse estudo foi possível apresentar o risco existente em sistemas desatualizados, que permitem o acesso e exploração por pessoas mal-intencionadas. Após o estudo, podemos concluir que existe uma grande necessidade de atenção por parte dos analistas envolvidos nos sistemas e infraestrutura de uma organização a estarem sempre atentos aos *paths* de atualizações, pois falhas que permitem a injeção de códigos maliciosos podem trazer prejuízos incalculáveis para uma empresa, como: perda de dados sigilosos, criptografia de arquivos, vetores de ataques, acessos bancários e danos à imagem da organização atacada.

Palavras-chaves: Segurança da informação, SMB, MS17-010, Eternalblue/DoublePulsar.

INTRODUÇÃO

Vivemos no século considerado o "século da informação", aos poucos o valor das empresas deixará de ser seus produtos e itens estocados para ser os seus arquivos, banco de dados, códigos fontes, dentre outros. Devido à importância dos dados como ativos, as empresas passaram a dar maior foco na área de TI (Tecnologia da Informação), criando novos cargos, ampliando e qualificando suas equipes, onde antes tinha um técnico de informática, hoje temos analistas de infraestrutura, de segurança da informação, DBA (*Data Base Administrator*), tudo isso porque os ladrões (atacantes) também evoluíram. Antes com uma arma branca ou de fogo, hoje são Crackers, especialistas em tecnologia com capacidade intelectual para invadir sistemas e banco de dados com o objetivo de capturar dados sigilosos, destruir sistemas ou vender informações no mercado negro.

De acordo com Galvão (2015):

> "A palavra cracker vem do termo *cracking*, que significa quebra. Os crackers são indivíduos que praticam a quebra de segurança de um sistema, ou seja, cometem delitos ou crimes digitais. Como os crackers cometem atos ilegais, são vistos como criminosos[...]"

De 2006 em diante, o número de incidentes com *cyber* ataques apresentou substancialmente crescimento, tornando-se um negócio rentável e a cada dia surgem novos indivíduos com o objetivo de lucrar de forma antiética e criminosa. Segundo a CERT.br (Grupo de Resposta a Incidentes de Segurança para a Internet no Brasil) somente no ano de 2019, ocorreram 875.327 incidentes reportados, 29,38% a mais em relação ao ano de 2018, conforme apresenta a Gráfico 1 em milhares de ataques.

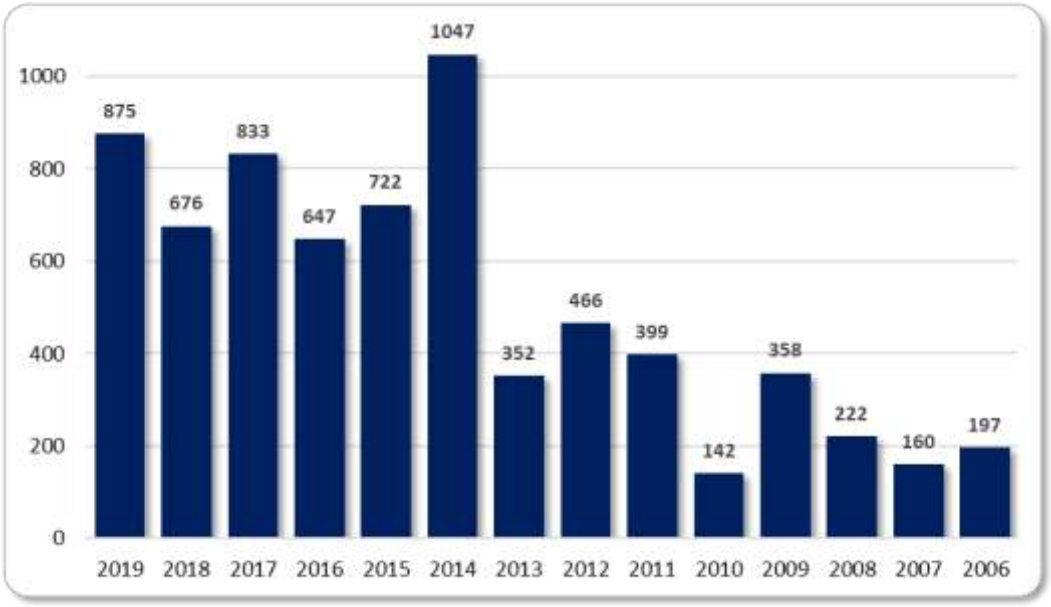

Gráfico 1 – Incidentes reportados ao CERT.br em milhares.
Fonte – https://cert.br/stats/incidentes/

De forma geral, pode-se dizer que um ataque cracker pode ocorrer de duas maneiras: falhas de sistemas e falhas humanas. "Nenhum sistema é 100% seguro", essa é uma frase muito conhecida pelos especialistas de segurança, é uma das afirmativas existentes na área de segurança, entretanto também vale ressaltar que de acordo com a USENIX (Associação norte-americana de profissionais de TI), 98% das pessoas não atualizam seus sistemas. Por exemplo, a falha no protocolo SMBv1 e SMBv2, também conhecida pelo nome de MS17-

010 apresentada neste artigo, foi corrigida 01 (um) mês antes da sua propagação acontecer, e 03 (três) anos após o ocorrido, a falha continua sendo explorada.

A MS17-010 é uma falha crítica anunciada pela Microsoft em 14 de março de 2017, no qual um atacante, ao explorar essa vulnerabilidade, consegue executar códigos maliciosos utilizando o protocolo SMBv1 e SMBv2 de compartilhamento de dados do sistema operacional Microsoft Windows, permitindo que o atacante tenha controle do hardware e software do computador alvo.

Este artigo tem por objetivo registrar a falha apresentada em uma empresa e todo o processo de exploração para a vulnerabilidade, assim como seus riscos e métodos de correções que foram realizados para sanar este problema.

REFERENCIAL TEÓRICO

PARROT OS

O ParrotOS é uma distribuição GNU/Linux baseado no Debian que possui foco na área de segurança da informação, projetado especialmente para testes de intrusão, análise forense em computadores, engenharia reversa, ataques em criptografias, dentre outros. Seu lançamento ocorreu em 10 de abril de 2013, sendo uma alternativa ao conhecido Kali Linux.

Uma das vantagens de se utilizar um SO voltado para a área de segurança está relacionado ao arsenal de ferramentas que comumente acompanham, sendo um conjunto de softwares para diversos campos da área de segurança da informação, como scanners de vulnerabilidade, *sniffers* de rede, ferramentas para análise e *brute force* em criptografias, testes de vulnerabilidades em redes wireless, banco de dados e várias outras ferramentas para todo o segmento, atualmente contando com mais de 500 softwares.

EXPLOIT

Um *exploit* é uma sequência de comandos, dados ou um software, produzido para explorar a vulnerabilidade de um sistema computacional, permitindo o acesso não autorizado, execução de comandos ou programas e até mesmo o controle total do sistema a qual foi implementado.

Muitos profissionais de segurança consideram os *exploits* um dos problemas mais sérios para os sistemas, principalmente nos casos de códigos desenvolvidos para vulnerabilidades somente descobertas por *blackhats* (*crackers*), essas falhas de segurança são chamadas de 0-day, e usadas abusivamente para exploração de falhas de segurança.

Segundo Perez e Santos (2013, p. 11):

> "O *exploit* é a ferramenta que através de linhas de comandos, pode acessar vulnerabilidades de sistemas operacionais, sites e programas, muitas vezes usados por pessoas mal-intencionadas que se aproveitam para roubar informações[...]".

Os *exploits* podem ser divididos em dois tipos, os conhecidos e os desconhecidos (*0-day*). Os *exploits* conhecidos, são aqueles que exploram vulnerabilidades que já foram publicadas, são os mais presentes nas notícias sobre segurança, e a qual os analistas podem tomar providências, evitando assim que os sistemas sejam atacados.

Por outro lado, como já mencionado existem os *exploits 0-day*, que são aqueles que exploram as falhas de segurança encontradas por crackers, e por este motivo não existem

medidas protetivas, podendo representar uma grave ameaça para a organização que estiver sendo direcionado, sendo considerado uma ameaça indetectável.

Neste artigo será apresentado a utilização de um *exploit* já conhecido como EternalBlue/DoublePulsar, utilizado para explorar e injetar códigos maliciosos através da falha de segurança contido no protocolo de compartilhamento de arquivos SMBv1 e SMBv2 presente no sistema Microsoft Windows 7.

PAYLOAD

Em segurança da informação, o *payload* é o código malicioso que executa uma ação dentro do sistema que foi anteriormente explorado por um *exploit*, podendo elevar privilégios, iniciar um processo, realizar uma conexão, acessar o *shell* do sistema e até recursos mais avançados, como um *meterpreter* (framework com diversas ferramentas já programas como: ligar a webcam, ativar um *keylogger*, tirar fotos da tela, capturar imagens da webcam, entre outros), também é possível executar um VNC *Injection*, permitindo controle total do computador na interface gráfica.

Um *payload* é um código injetado para que o sistema execute e que seja entregue pelo *Metasploit*. Por exemplo, um *reverse shell* é um *payload* que cria uma conexão reversa da máquina alvo de volta para a máquina do atacante. Um *payload* também pode ser algo simples como alguns comandos a serem executados no sistema operacional alvo (KENNEDY, 2011).

Sistemas operacionais voltados para a área de segurança possuem mecanismos para auxiliar no processo de construção de um *payload*, como por exemplo a Figura 1 abaixo, que demonstra a construção com a ferramenta *msfvenom*:

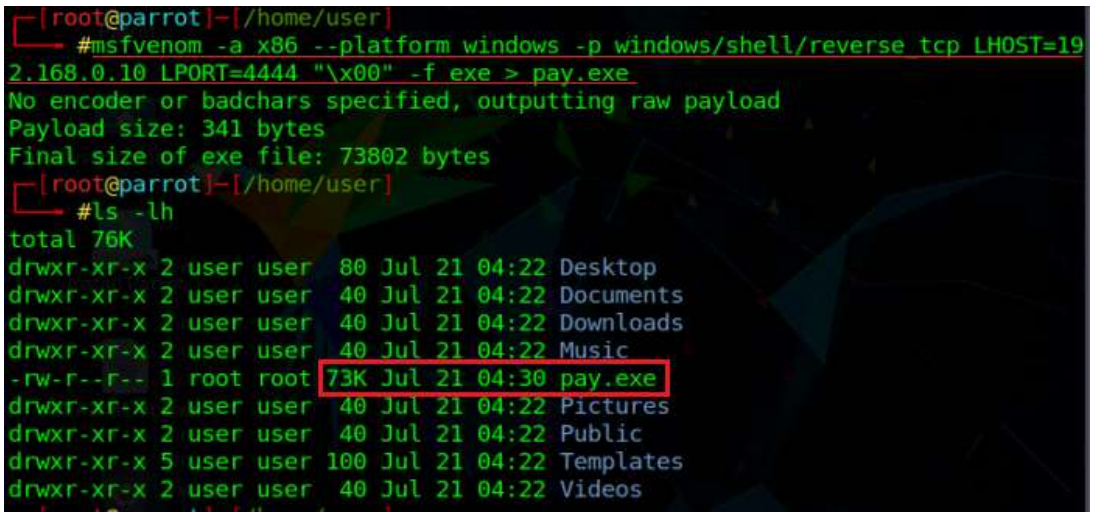

Figura 1 – Construção de Payload com msvenom.
Fonte – Autoria própria.

Como é possível ver na Figura 1, é executado o comando para a construção de um simples *payload* para realizar uma conexão reversa a um Host com o IP 192.168.0.10 na porta 4444, também são definidos alguns parâmetros como a arquitetura do alvo (-a x86) e para qual plataforma o *payload* está sendo construído (*--platform* Windows). Após a execução é gerado um arquivo com o nome definido na execução de pay.exe contendo o código de conexão reversa.

NMAP

Segundo nmap.org

> O Nmap (“*Network Mapper*”) é uma ferramenta de código aberto para exploração de rede e auditoria de segurança. Ela foi desenhada para escanear rapidamente redes amplas, embora também funcione muito bem contra hosts individuais. O Nmap utiliza pacotes IP em estado bruto (*raw*) de maneira inovadora para determinar quais hosts estão disponíveis na rede, quais serviços (nome da aplicação e versão) os hosts oferecem, quais sistemas operacionais (e versões de SO) eles estão executando, que tipos de filtro de pacotes/firewalls estão em uso, e várias outras características.

O Nmap é uma ferramenta para realizar varredura nos sistemas operacionais, podendo ser configurada para análise de uma porta específica ou de todo um conjunto, também para um computador ou para toda uma *range* de IPs da rede.

Para muitos o Nmap é apenas um programa para varredura de portas na rede de computadores, entretanto a utilização desse software pode ir de um simples scanner para verificação de uma porta aberta, como também a execução de *scripts* para verificação de vulnerabilidades, reconhecimento de sistema operacional, serviços que estão rodando no computador, como também suas versões, possuindo várias outras características.

A Figura 2 abaixo, mostra uma utilização simples e explicativa do Nmap:

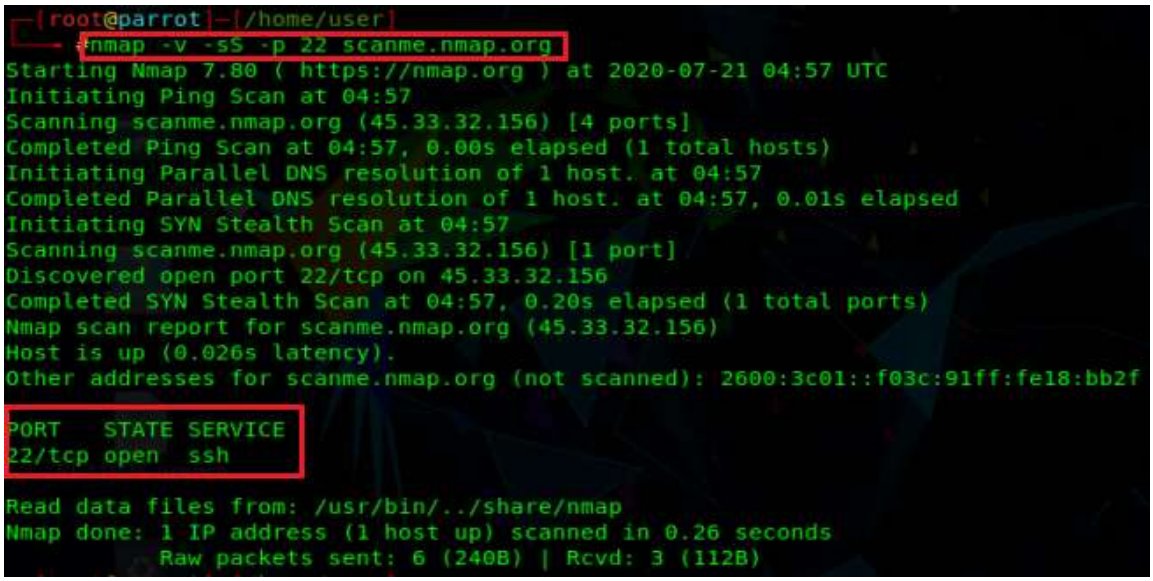

Figura 2 – Varredura com nmap.
Fonte – Autoria própria.

-v : Modo *Verbose*, permite acompanhar cada etapa do processo.

-sS : Chamado de *Scan* SYN, opção padrão e mais popular, ele é não-obstrusivo e camuflado pois nunca finaliza uma conexão TCP.

-p ; Especifica uma porta para análise, caso o parâmetro não seja inserido, serão analisadas as 1000 (mil) principais portas do sistema.

Scanme.nmap.org : DNS para varredura, também podendo ser utilizado IP.

O modelo apresentado acima é a estrutura mais básica a ser utilizada, existem diversos parâmetros para utilização em cada caso específico de necessidade. Para conhecer mais detalhes, acessar o site da documentação em https://nmap.org/man/pt_BR/.

SMB

Server Message Block (SMB) é um protocolo para compartilhamento de arquivos em rede, permitindo que computadores realizem a leitura e gravação de dados em uma rede, e extremamente utilizado nos computadores com o sistema operacional Microsoft Windows.

O protocolo trabalha no designer de cliente/servidor, o cliente envia requisições e o servidor fornece as respostas, essa comunicação ocorre via pacotes e cada pacote contém um cabeçalho padrão e mais dois campos de tamanhos variáveis utilizados para informações específicas da comunicação. No cabeçalho também acompanha um campo chamado "*command*", esse campo indica o propósito do pacote, como por exemplo se é uma requisição de *login*, leitura de arquivos, escrita ou abertura.

A Figura 3 abaixo, apresenta o *template* de um cabeçalho do protocolo SMB:

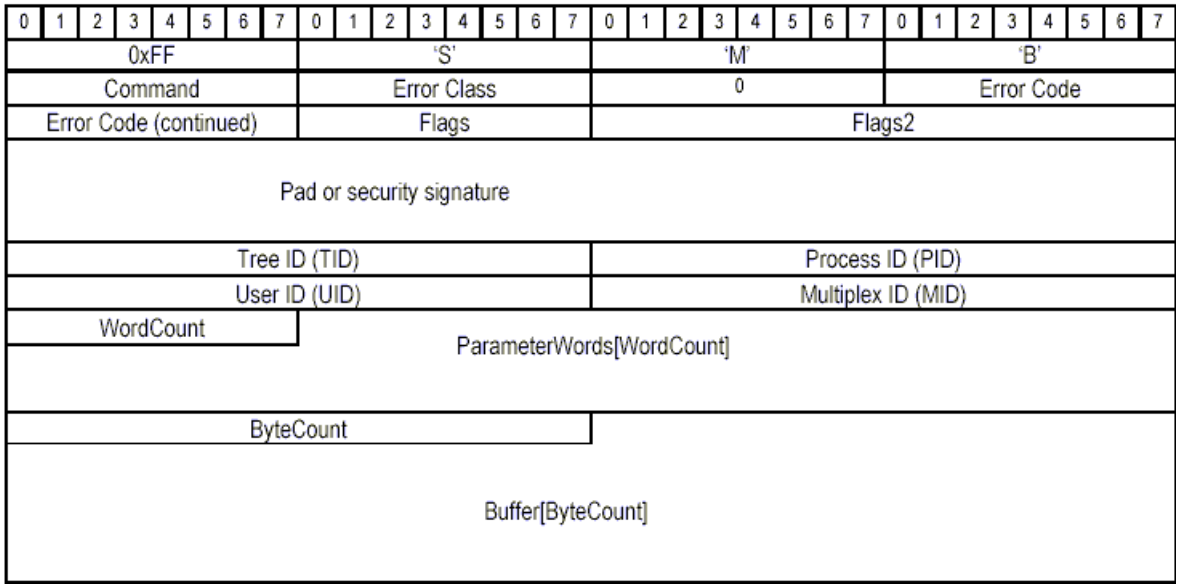

Figura 3 – Cabeçalho SMB.
Fonte – https://www.gta.ufrj.br/grad/01_2/samba/smbcifsinternamente.htm.

Cada pacote SMB possui a mesma estrutura, a primeira linha campo 4 octetos (4 blocos de 8 bits) contendo o valor *0xFF* e seguidos pelas letras 'S', 'M' e 'B', segue a explicação de cada campo do cabeçalho:

Command : Contendo um espaço de 8 bits para identificação do objetivo do pacote, se é uma abertura, leitura ou escrita de arquivo ou requisição de *login*. O protocolo SMBv1 possui 100 (cem) comandos possíveis, seu antecessor SMBv2 teve uma redução para apenas 19 (dezenove) comandos, por exemplo: SMB_COM_READ_ANDX7 (0x2e), SMB_COM_TREE_CONNECT (0x70) e SMB_COM_NEGOTIATE (0x72).

Error Class : Código representando se houve sucesso ou um erro na requisição, normalmente enviada com código "0" (zero) para indicar que foi uma requisição bem sucedida, porém quando acompanhada de valores diferentes de "0" (zero), então deve-se analisar a qual classe de erro o código se refere, seguem exemplos classes de erros:

- ERRDOS (0x01) – Erro do núcleo do conjunto de instruções do sistema operacional DOS.
- ERRSRV (0x02) – Erro gerado pelo gerenciador de arquivos de rede do servidor.
- ERRHRD (0x03) – Erro no hardware.
- ERRCMD (0xFF) – O comando não estava no formato 'SMB'.

Error Code : Possuindo 18 bits de tamanho, é utilizado para identificar o erro que ocorreu, quando em conjunto com o *error class* é possível identificar o erro ocorrido, comumente identificado com o valor "0" (zero) para representar que não houve falha na requisição. Assim como o *error class*, esse campo é inserido pelo servidor para resposta da requisição solicitada.

Flags e Flags2 : Quase todos os bits desses octetos representam opções particulares, em exceção de alguns específicos com o bit 3 do octeto *Flags* que é utilizado para informar que tudo deve ser tratado sem preocupação com *case sensitive.* Também o bit 6 do octeto *Flags2* que indica que qualquer caminho da requisição pode ser um arquivo com nome longo. Entre outras opções que não será abordado neste artigo.

Pad/Security signature : Comumente inserido com o valor "0" (zero).

Tree ID (TID) : É um campo de 16 bits que identifica a qual recurso o pacote está se referindo, se é um compartilhamento de arquivo ou impressora por exemplo, e quando é realizado a troca de pacotes que não tem relação com um recurso, o valor que acompanha não faz sentido e é ignorado. Quando um cliente quer acesso a um recurso, ele deve enviar um pacote com o campo de comando *SMB_COM_TREE_CONNECT_ANDX*, neste pacote o nome do compartilhamento ou da impressora é especificado (\\Server\Dir), o servidor vai verificar se o recurso existe e se o cliente tem acesso, feito isso, retorna com a resposta indicando o sucesso ou não.

Process ID (PID) : Utilizado para verificar que processo está fazendo a requisição, muito utilizado pelo servidor para evitar que os arquivos não serão corrompidos por processos concorrentes.

User ID (UID) : Após realizar a requisição de *login* no servidor é gerado um UID para identificação do usuário que está fazendo as requisições.

Multiplex ID (MID) : Utilizado para controlar múltiplas requisições, ao enviar uma requisição o servidor verifica se já existe alguma requisição pendente.

WordCount e parameter words : Os pacotes utilizam estes campos para armazenar os dados de comandos. O *template* apresentado acima não suporta todos os possíveis tipos de dados do protocolo SMB, por este motivo o *parameter words* possui um tamanho variável e o *wordcount* possui a quantidade de palavras que o *parameter words* irá conter, desta forma ele será ajustado conforme a necessidade.

ByteCount e buffer : Este campo funciona de forma similar aos dois acima descritos, o campo *bytecount* contém a quantidade de bytes que será utilizado no campo *buffer*. A principal diferença entre buffer e *parameter words* é o tipo de dado que cada um armazena.

METODOLOGIA

Este projeto tem características de uma pesquisa aplicada, em que os conceitos estudados são desenvolvidos visando a utilização no mundo real. Pesquisa aplicada é voltada à absolvição de conhecimentos com o uso da aplicação proposta numa situação específica (GIL, 2010).

Pesquisa aplicada tem como objetivo provocar conhecimento para a aplicação prática conduzidos à resolução de problemas específicos relaciona verdades e interesses locais (SILVA, 2001).

A pesquisa apresentada neste artigo é um estudo de caso real de teste de intrusão na rede de computadores de uma empresa após constatação de uma falha grave de segurança. Por questões éticas de pesquisa, o nome da empresa e das pessoas envolvidas serão omitidas. Para dar sustentação teórica a pesquisa de campo, foi utilizada também utilizada a pesquisa bibliográfica, constituída de acesso a livros, artigos científicos, teses e dissertações.

O processo de *PenTest* foi realizado utilizando o sistema operacional *Parrot Security OS* que não é tão conhecido como o popular Kali Linux, mas também possui grande arsenal de ferramentas para análise e exploração de falhas web, desktop, mobile e redes de computadores. No ataque descrito nos próximos tópicos foram utilizadas as ferramentas Nmap para análise, identificação e varredura dos computadores da rede e o framework *msfconsole* contendo todo o suporte para análise e exploração de falhas na segurança, além de comandos básicos do Linux visando maior controle e organização na apresentação dos dados.

Para melhor compreensão da metodologia aplicada para exploração, a pesquisa foi subdivida nas seguintes etapas: varredura, análise de vulnerabilidade e exploração.

VARREDURA

A primeira etapa para identificação é a varredura da rede, cujo objetivo é a busca de computadores que possuem o sistema de compartilhamento do Windows ativo. Sabendo que este serviço funciona através da porta 445, foi utilizado a ferramenta *Nmap* para pesquisar por toda uma *range* de IP em busca de computadores com este serviço rodando.

A Figura 4 abaixo, mostra a utilização e apresentação dos dados.

Figura 4 – Varredura com nmap.
Fonte – Autoria própria.

Conforme apresentado na Figura 5 acima, foi utilizado o comando "**nmap -v –p 445 <ip_do_host> -oG <nome_do_arquivo>**". Segue abaixo a explicação desta execução:

-v: Modo verbose, permite acompanhar na tela cada procedimento realizado pelo software.

-p 445: Define uma porta específica para realização do scanner, evitando perca de tempo na busca por outras portas.

-oG: Gera um arquivo no formato "*Grepable*" contendo as principais informações de resultado.

Após gerar o arquivo é possível no *Linux*, utilizando o comando "cut", filtrar os dados impressos, gerando uma listagem somente com os hosts que apresentaram o resultado de "open", sem nenhum filtro sendo aplicado por sistemas de firewall. A Figura 5 abaixo, mostra como foi impressa a nova lista, contendo somente os IPs.

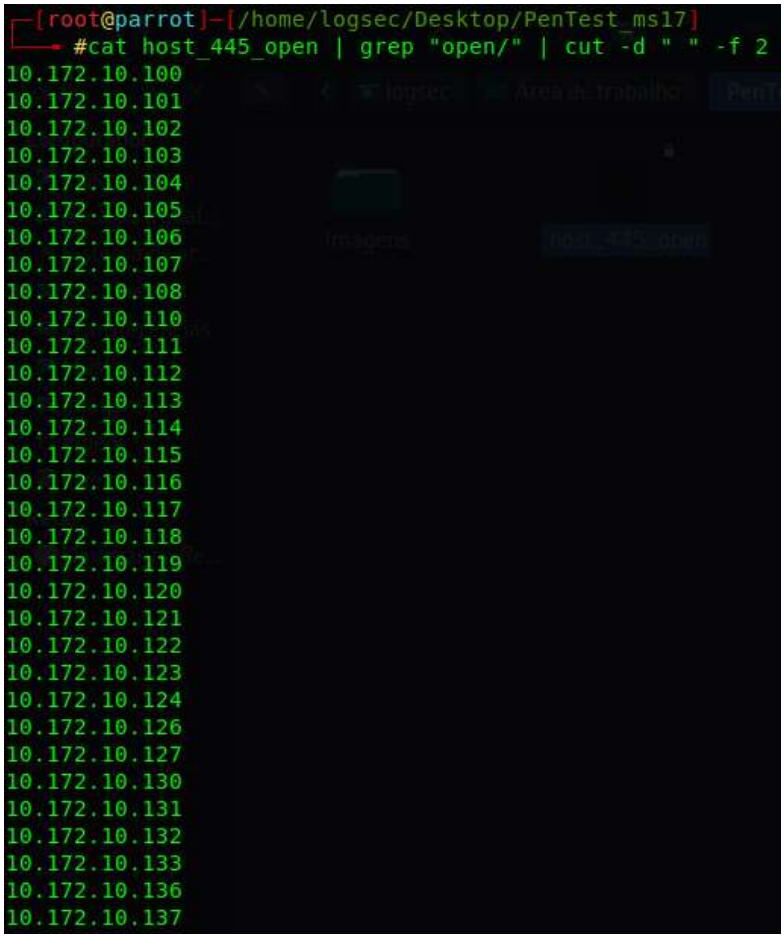

Figura 5 – Criando arquivo de Hosts.
Fonte – Autoria própria.

ANÁLISE DE VULNERABILIDADE

Agora, sabendo quais os hosts da rede possuem a porta aberta sem proteção, é possível utilizar um módulo auxiliar do Framework *Metasploit* para verificar se o serviço SMB que está rodando é vulnerável.

Para isso, é necessário acessar o módulo auxiliar chamado **smb_ms17_010** e configurá-lo para receber a listagem, conforme mostra a Figura 6 abaixo:

```
msf5 auxiliary(scanner/smb/smb_ms17_010) > options

Module options (auxiliary/scanner/smb/smb_ms17_010):

   Name         Current Setting
   ----         ---------------
   CHECK_ARCH   true
   CHECK_DOPU   true
   CHECK_PIPE   false
   NAMED_PIPES  /usr/share/metasploit-framework/data/wordlists/named_pipes.txt
   RHOSTS       file:/home/logsec/Desktop/PenTest_ms17/ips_445_open
   RPORT        445
   SMBDomain    .
   SMBPass
   SMBUser
   THREADS      1
```

Figura 6 – Preparando o auxiliar para análise de vulnerabilidade.
Fonte – Autoria própria.

Vários parâmetros são definidos, mas vale ressaltar o parâmetro RHOST, que contém o caminho do arquivo com a lista de hosts para testar e o parâmetro RPORT, onde é especificada a porta para teste.

Após essa execução, o auxiliar irá se encarregar de testar IP por IP em busca da vulnerabilidade, apresentando a mensagem se existe ou não a falha de segurança. A Figura 7 abaixo, demonstra o resultado do módulo:

```
┌─[root@parrot]─[/home/logsec/Desktop/PenTest_ms17]
└──╼ #cat host_445_open | grep "open/" | cut -d " " -f 2
10.172.10.100
10.172.10.101
10.172.10.102
10.172.10.103
10.172.10.104
10.172.10.105
10.172.10.106
10.172.10.107
10.172.10.108
10.172.10.110
10.172.10.111
10.172.10.112
10.172.10.113
10.172.10.114
10.172.10.115
10.172.10.116
10.172.10.117
10.172.10.118
10.172.10.119
10.172.10.120
10.172.10.121
10.172.10.122
10.172.10.123
10.172.10.124
10.172.10.126
10.172.10.127
10.172.10.130
10.172.10.131
10.172.10.132
10.172.10.133
10.172.10.136
10.172.10.137
```

Figura7 – Retorno das máquinas vulneráveis.
Fonte – Autoria própria.

O resultado informado retorna que o host é vulnerável e traz as informações específicas do sistema operacional em que foi realizado o teste, essas informações são extremamente importantes para o preparo do software de exploração, chamado de "*Eternalblue"* e "*Doublepulsar*". O próximo tópico apresentará a configuração e utilização do sistema para exploração.

EXPLORAÇÃO

Para o processo de exploração foi necessário o levantamento das informações específicas da máquina alvo, este procedimento é importante para a configuração do *exploit*, pois para o sucesso é necessário que ele seja configurado com as exatas informações de arquitetura, versão, processo a ser explorado, dentre outras informações.

A exploração é realizada utilizando dois métodos de ataque conhecidos como *EternalBlue* e *DoublePulsar*, métodos vazados da *NSA (National Security Agency)*. A Figura 8 mostra a preparação do *exploit* para realizar a exploração do alvo.

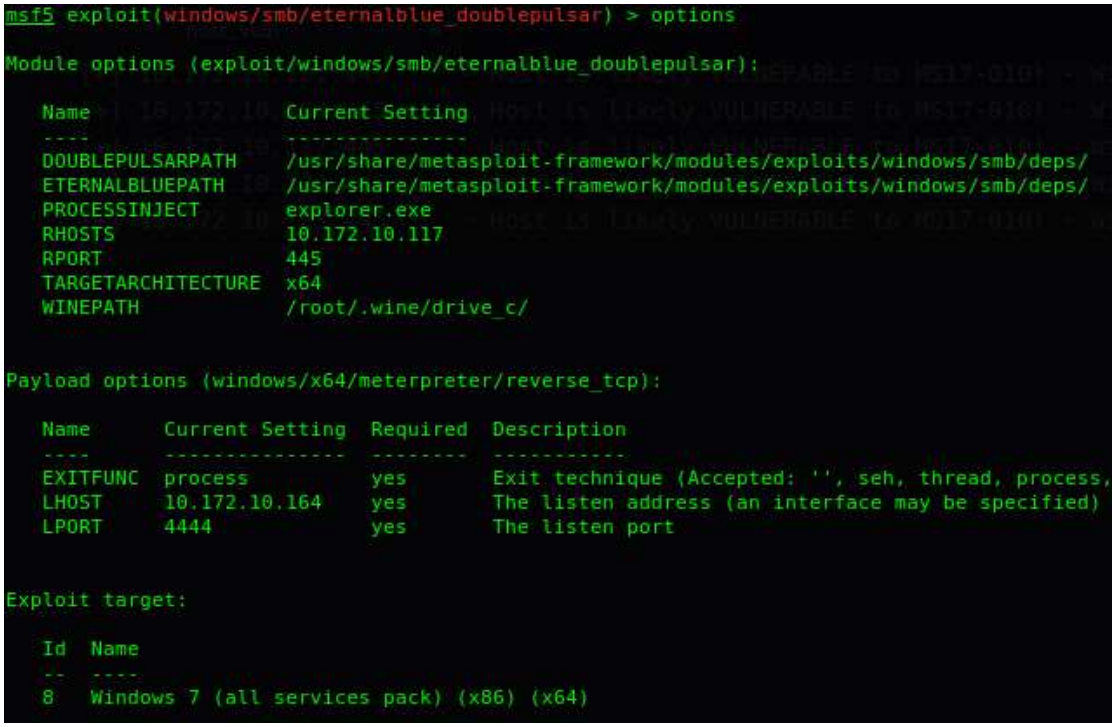

Figura 8 – Preparação do exploit.
Fonte – Autoria própria.

Para um entendimento breve do descrito na Figura 9, serão apresentados alguns parâmetros e suas funções:

PROCESSINJECT: Processo que será explorado na máquina alvo, que no caso acima, será utilizado o explorer.exe.

RHOST: IP da máquina alvo.

RPORT: Porta do protocolo SMBv1 que será explorada.

TARGETARCHITECTURE: Arquitetura do sistema operacional alvo (x86, x64).

TARGET: Sistema operacional a ser atacado, podendo sofrer variação de arquitetura ou *service pack* do S.O.

Além destas configurações, também é necessário carregar o *payload*, este arquivo é o que carrega o código malicioso, que no caso acima foi utilizado um contendo com código para conexão reversa. Este tipo de *payload* vai abrir uma porta na máquina do atacante, e ao conseguir inserir o arquivo de código malicioso na máquina alvo, vai retornar com a conexão e desta forma vai liberar a comunicação entre os dois computadores.

Ao executar o *exploit*, será explorada a falha de execução de código malicioso através do protocolo de compartilhamento do Windows, permitindo então que seja feita uma conexão reversa, permitindo que seja executado os scripts do "*Meterpreter*", conforme apresenta a Figura 9.

Figura 9 – Ganho de acesso com Meterpreter.
Fonte – Autoria própria.

Um acesso do tipo *Meterpreter* é considerando uma das formas mais avançadas de exploração, pois o *Meterpreter* carrega um conjunto de ferramentas já configuradas, permitindo que sejam executados vários tipos de ações. Seguem alguns dos comandos permitidos:

- **PS**: Lista de processos que o Windows está executando naquele exato momento;
- **MIGRATE** + número do processo: Migra para um processo que está sendo executado Exemplo: *migrate* 1197;
- **SYSINFO**: Mostra qual a versão do Windows da máquina alvo;
- **GETSYSTEM**: Eleva o nível de privilégio para *SYSTEM;*
- **IPCONFIG**: Mostra as configurações de rede da máquina alvo;
- **SCREENSHOT**: Salva um arquivo JPEG com um *print* da tela alvo;
- **KEYSCAN_START**: Inicia o *keylogger*;
- **KEYSCAN_DUMP**: Visualiza os dados capturados após o *keyscan_start*;
- **KEYSCAN_STOP**: Para a captura do teclado;
- **RUN PERSISTENCE –X**: Esse comando configura a máquina alvo para que a cada *boot* no sistema ela estabeleça novamente a conexão com a máquina atacante;
- **HASHDUMP**: Faz a captura do *hash* de senhas do computador alvo;
- **EXECUTE**: Executa um aplicativo. Exemplo: **execute -f cmd.exe**;
- **SHELL**: Abre um *prompt* de comando;
- **CLEAREV**: Limpa os logs de eventos do Windows.

Utilizando o comando *shell* do *meterpreter*, será transferido para a máquina do atacante o *prompt* de comando do alvo, podendo gerenciar o sistema operacional com as mesmas permissões caso estivesse fisicamente no computador atacado. Caso o atacante queira, é possível iniciar processos ou encerrá-los, verificar configurações da máquina e da rede em que está presente, ou utilizar este computador como um vetor de ataque a outros computadores da rede, utilizando-o como uma máquina "zumbi" para evitar rastreio, após infectado são várias as possibilidades de execução.

Na Figura 10 abaixo, mostra o acesso ao *prompt* da máquina alvo em que foi visualizado as configurações de rede, domínio e nome do computador que foram omitidas por questões éticas.

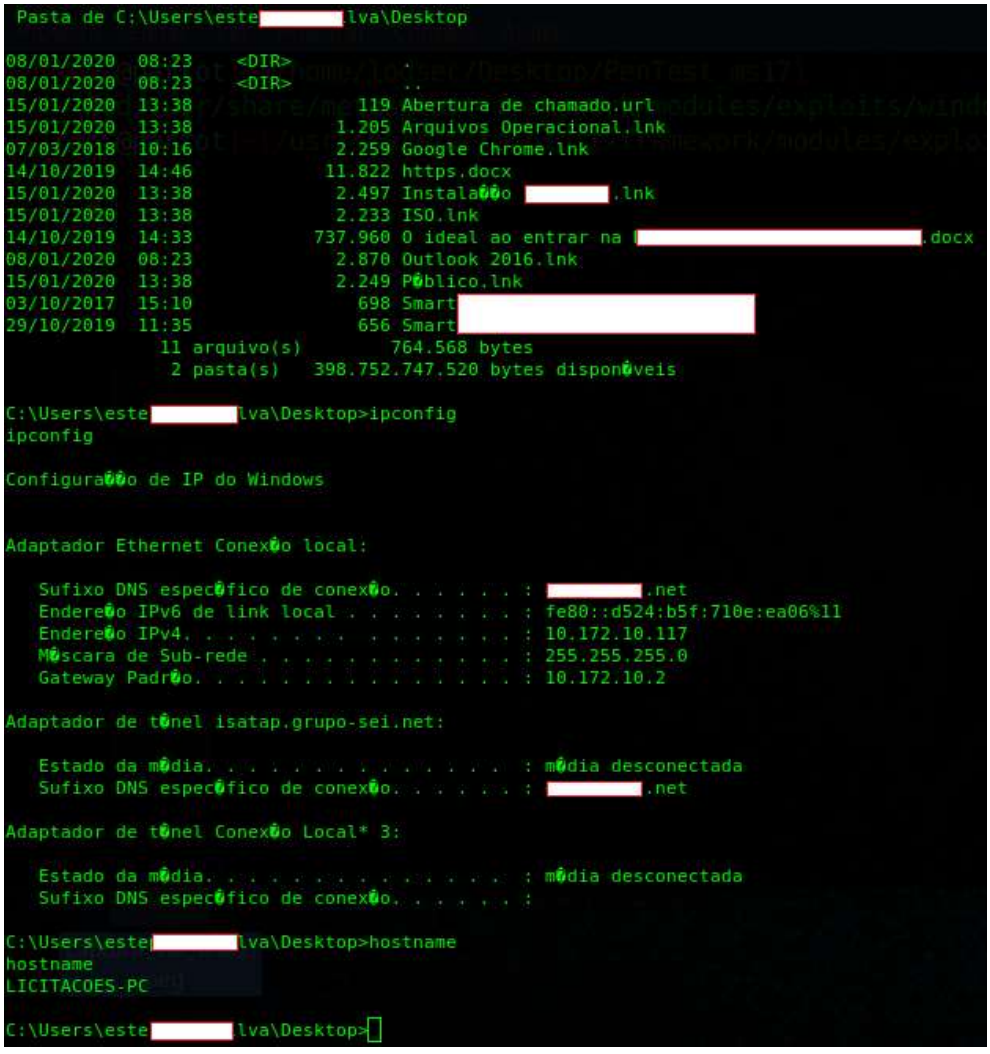

Figura 10 – Acesso ao prompt de comando da máquina alvo.
Fonte – Autoria própria.

Conforme descrito acima, o *Meterpreter* também permite a captura de telas com o comando "*screenshot*", registrando uma imagem em tempo real do computador infectado, a foto é tirada sem apresentar nenhum efeito que possa ser identificado pelo alvo, que continua utilizando a máquina sem saber que está sendo invadido e monitorado.

A Figura 11 abaixo, mostra uma captura enquanto o usuário utilizava o computador:

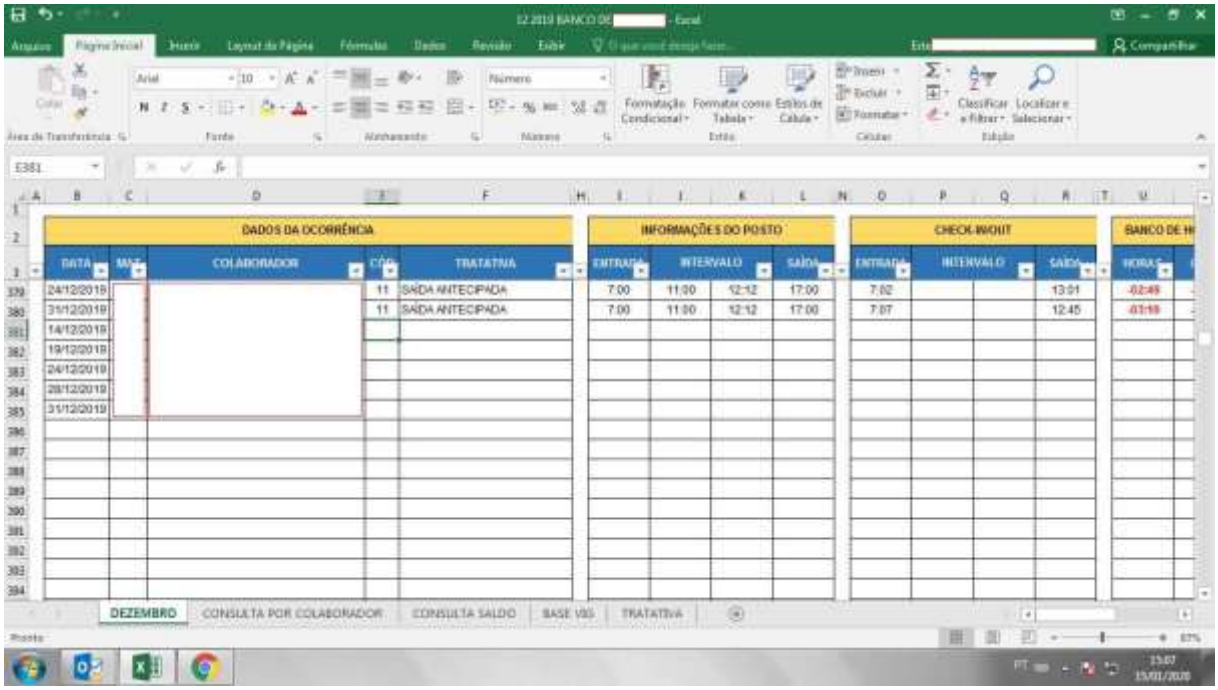

Figura 11 – Screenshot da tela do alvo.
Fonte – Autoria própria.

Para finalizar a demonstração do *Meterpreter*, foi possível inserir um código malicioso para compartilhamento da tela em tempo real enquanto era utilizada, isso é possível realizando um ataque de VNC *Injection*, o VNC é um sistema de compartilhamento gráfico, que utiliza o protocolo RFB (*Remote Frame Buffer*) e permite que o computador seja controlado remotamente.

A Figura 12 a seguir, é uma foto da utilização que estava ocorrendo em tempo real pelo alvo do ataque.

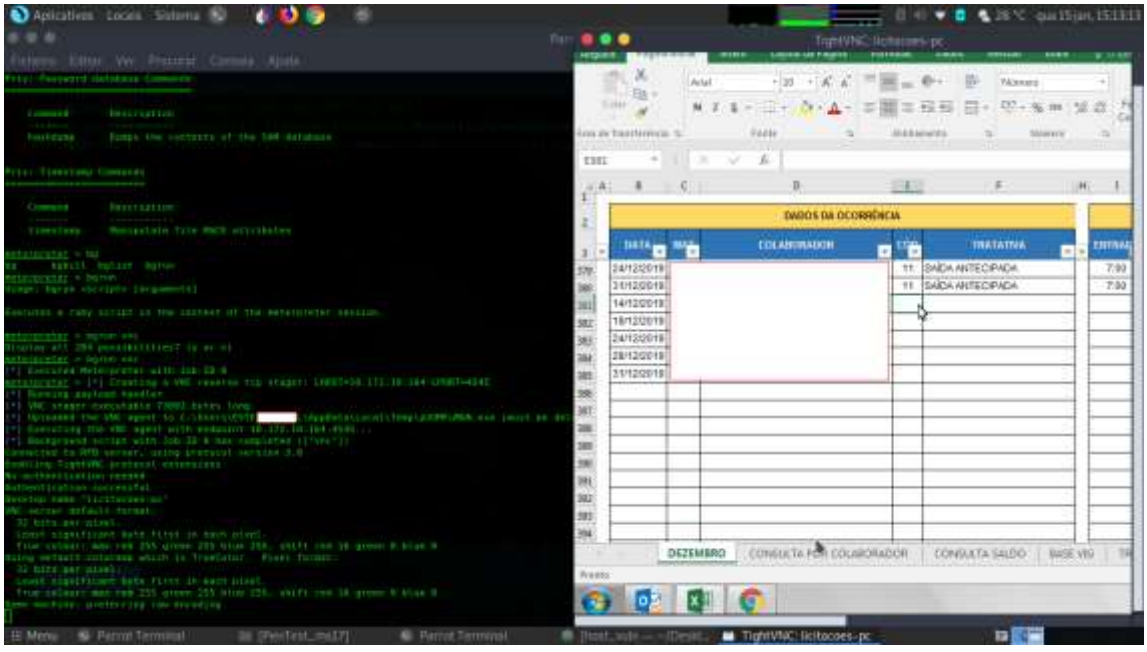

Figura 12 – Foto da captura da tela em tempo real.
Fonte – Autoria própria.

CONCLUSÕES

A falha MS17-010 é considerada uma das mais críticas existentes da Microsoft, falha esta que foi explorada pelo Ransomware WannaCry em maio de 2017, sendo o motivo deste malware ter se espalhado pela rede de computadores do mundo inteiro.

Para corrigir essa vulnerabilidade a Microsoft lançou o Boletim de segurança contendo a atualização 4012598, que de fato resolve este problema para computadores Windows 7 em diante para a falha do SMBv1 e v2.

As vulnerabilidades sempre vão existir, mas é preciso estar atualizado com as notícias de segurança e novas falhas/exploits lançadas no mercado, essa preocupação precisa partir principalmente dos analistas responsáveis e a pesquisa no campo de segurança precisa estar em constante inovação e evolução. A segurança computacional precisa ser atualizada com novos métodos de acesso e modelos atualizados de comunicação.

Atualmente vivemos uma nova realidade com o surgimento da computação quântica que possui processamento muito superior a um supercomputador de hoje, e por este motivo, especialistas de segurança já devem se preocupar com sistemas de "descriptografias" ou "*brute force"*, processos que atualmente precisam ser muito bem estudados e trabalhados para apresentarem algum resultado, entretanto com a computação quântica, será possível quebrar senhas rapidamente.

O artigo apresentado, além de demonstrar a falha de segurança e sua correção, também chama a atenção dos usuários e especialistas para a necessidade de manter os softwares constantemente atualizados, pois algumas pessoas possuem ainda são resistentes com esses procedimentos, onde o nível de segurança do sistema é negociado com a usabilidade.

Segundo Besnard e Arief (2003, 259)

> [...] a segurança é descrita como um processo de várias camadas em que uma variedade de usuários (desenvolvedores, analistas de segurança, usuários finais.) Possuem um papel a desempenhar. Cada um desses usuários tem impacto na segurança da informação. Administradores e/ou usuários finais, por exemplo, ao não realizar as atualizações para os softwares de antivírus, deixam brechas para ataques.

RECOMENDAÇÕES

Para qualquer sistema a recomendação é sempre utilizar a sua versão mais nova, para que as falhas de segurança que foram identificadas sejam tratadas. Como este artigo tratou especificamente do protocolo de compartilhamento de arquivos do Windows (SMBv1 e SMBv2), é altamente recomendável que não seja mais utilizado o sistema operacional Windows 7, pois em 14 de janeiro de 2020 a Microsoft encerra o suporte ao sistema operacional, isso significa que se surgir mais alguma falha de segurança ou de utilização, a Microsoft não tratará mais.

Se não for possível realizar a migração do Windows 7 para Windows 10, como já mencionado neste artigo, a atualização foi disponibilizada no site da Microsoft para download, resolvendo o problema desta vulnerabilidade específica. Para o teste realizado neste artigo, após a identificação e exploração da falha, foi instalado em todos os computadores o *path* de atualização 4012598, e novamente realizado o processo de

escaneamento da vulnerabilidade na porta 445 para confirmar se todos os computadores estavam livres da vulnerabilidade mencionada neste artigo.

Realizado um novo *scanner* da vulnerabilidade após aplicação da atualização, conforme mostra a Figura 13, não é mais apresentada a falha na segurança.

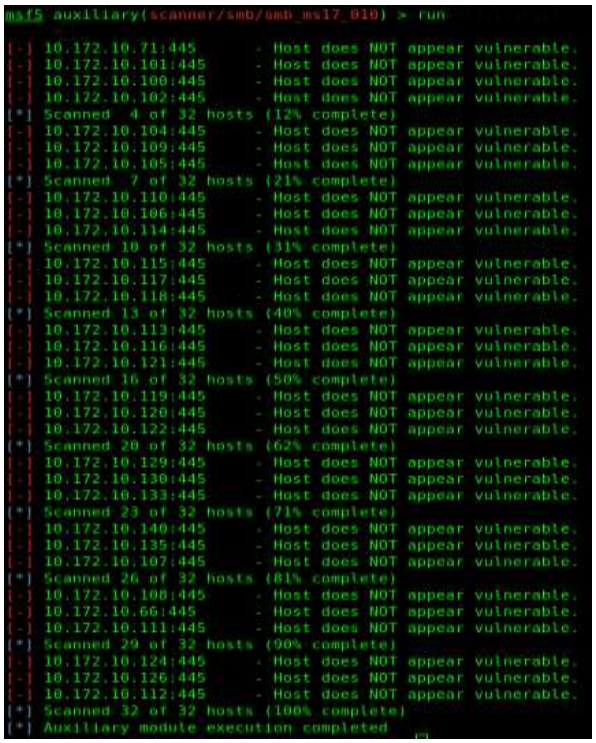

Figura 13 – Resultado do scanner após correção.
Fonte – Autoria própria.

Segue abaixo o links para download do *Path*

https://docs.microsoft.com/pt-br/security-updates/SecurityBulletins/2017/ms17-010?redirectedfrom=MSDN

Também é valido ressaltar que a vulnerabilidade apresentada neste artigo afeta o protocolo SMBv1 e v2, que foram desenvolvidos respectivamente para o Windows 7 e Vista. A partir do Windows 10, o sistema operacional da Microsoft passou a utilizar o protocolo SMBv3 que não possui falha para exploração com EternalBlue/DoublePulsar, entretanto já foi anunciado pela Microsoft a necessidade de remover a compressão desse novo protocolo, pois ele também possui uma falha de segurança.

É possível verificar pelo descrito acima que novas falhas de segurança vão surgindo a cada dia, é de responsabilidade do analista de segurança ou infraestrutura estar sempre atualizado com as novas ameaças e falhas que estão surgindo nos fóruns de discussões, desta forma precavendo e mitigando os riscos de exploração de uma falha já conhecida e reportada pela fabricando do software.

REFERÊNCIAS BIBLIOGRÁFICAS

CODEFX. CIFS Explained. 2001. Disponível em http://www.codefx.com/CIFS_Explained.htm

BESNARD, Denis; ARIEF, Budi. Computer security impaired by legitimate users. United Kingdom. 2003.

DANTAS, Marcus Leal. Segurança da informação: uma abordagem focada em gestão de riscos. Olinda: Livro Rápido, 2011.

ENGEBRETSON, Patrick. Introdução ao Hacking e aos Testes de Invasão: facilitando o hacking ético e os testes de invasão. São Paulo: Novatec, 2014.

ERICKSON, Jon. Hacking: The Art of Exploitation. Ed. 2. Nostarch: San Francisco, 2008.

GALVÃO, Michele C. Fundamentos em Segurança da Informação. Pearson: São Paulo, 2015.

GIL, A. C. Como Elaborar Projetos de Pesquisa. Atlas, 2010. v. 5ª ed.

GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO. SMB/CIFS internamente. 2020. Disponível em https://www.gta.ufrj.br/grad/01_2/samba/smbcifsinternamente.htm

HALL, Gary; WATSON, Erin. Hacking: Computer Hacking, Security Testing,Penetration Testing, and Basic SecurDec 28. CreateSpace Independent Publishing Platform, 2016.

HERTZOG, Raphael; O'GORMAN, Jim. Mastering the Penetration Testing DistributionJun. Ed. 1. OffSec Press: Cornelius, 2017.

KENNEDY, David. et al. Metasploit: the penetration tester's guide. San Francisco: No Starch Press, 2011

LOPES FILHO, Cesar G. et al. Análise de Vulnerabilidades em Redes de Computadores– Estudo de Caso com a Ferramenta Nmap.

LYON, Gordon Fyodor. Nmap network scanning: The official Nmap project guide to network discovery and security scanning. Insecure, 2009.

Nmap Network Scanning. Guia de Referência do Nmap (Página do Manual). 2020. Disponível em: <https://nmap.org/man/pt_BR/>

PEREZ, Ana Carolina; SANTOS, Ariadne Carolina Gomes. APLICAÇÃO DE MÉTRICAS DE SEGURANÇA NA PRENVEÇÃO DE ATAQUES DE "KERNEL EXPLOITATION" EM DISTRIBUIÇÃO LINUX. São Paulo, 2013.

RAMOS, Rodrigo. Um estudo do Nmap baseado em Kali Linux como ferramenta de apoio para a Computação Forense Preventiva.

SILVA, Edna Lúcia; MENEZES, Estera Muszkat. Metodologia da pesquisa e elaboração de dissertação. 3. ed. rev. atual. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2001. 121p.

VELU, Vijay K. Mastering Kali Linux for Advanced Penetration Testing - Second Edition: Secure your network with Kali Linux - the ultimate white hat hackers. Ed. 2. Birmingham, 2016.

WEIDMAN, Georgia. Testes de Invasão: uma introdução prática ao hacking. São Paulo: Novatec, 2014.