

Trabalho Prático

Backdoor Attack

Licenciatura em Eng. Informática
Segurança Informática



Helder Godinho 42741
Mariana Silva 54389
Docente: Pedro Patinho

1 Introdução

A segurança informática é uma área crítica no mundo atual, onde sistemas e redes são constantemente ameaçados por diversas formas de ataques. Este relatório documenta a criação de uma backdoor, uma técnica de ataque que permite a um invasor obter acesso não autorizado a um sistema.

O objetivo deste trabalho é explorar o funcionamento de uma backdoor e compreender as vulnerabilidades que ela explora, contribuindo para um melhor entendimento das medidas de segurança necessárias para proteger sistemas informáticos.

2 Contexto Teórico

2.1 Conceitos de Backdoor

Uma backdoor é um método utilizado para obter acesso remoto a um sistema ou rede, ignorando os mecanismos normais de autenticação. As backdoors são frequentemente utilizadas por invasores para manter um acesso persistente a sistemas comprometidos.

2.2 Métodos Comuns de Implantação de Backdoors

Backdoors podem ser implementadas de diversas maneiras, incluindo:

- Modificação de software legítimo para incluir código malicioso.
- Exploração de vulnerabilidades em aplicações ou sistemas operacionais.
- Utilização de engenharia social para convencer usuários a executar software malicioso.

2.3 Riscos e Consequências de Backdoors

As backdoors representam um grande risco de segurança, pois permitem que invasores controlem sistemas comprometidos, roubem dados sensíveis e causem danos. A detecção e remoção de backdoors pode ser complexa, e sua presença pode comprometer a integridade de toda a rede.

3 Descrição do Trabalho

3.1 Objetivo do Trabalho Prático

O objetivo deste trabalho prático consiste em desenvolver uma backdoor funcional e um servidor para controlar essa backdoor remotamente. Este trabalho permitiu-nos compreender como as backdoors operam e quais medidas podem ser implementadas para mitigá-las.

3.2 Ferramentas e Linguagens Utilizadas

- Linguagem de programação: Python
- Bibliotecas: `socket`, `subprocess`, `os`, `json`, `keyboard`, `ImageGrab`.

3.3 Descrição dos Arquivos

- `backdoor.py`: Contém o código da backdoor que será instalada na máquina alvo.

- `server.py`: Contém o código do servidor que comunica com a backdoor, enviando comandos e recebendo respostas.

4 Execução

4.1 Como Executar o Código

1. Configuração do Servidor:

- Execute o `server.py` na máquina que atuará como servidor.
- O servidor aguardará conexões na porta 5555.

2. Implantação da Backdoor:

- Execute o `backdoor.py` na máquina alvo.
- A backdoor tentará conectar-se ao servidor configurado.

4.2 Funções Implementadas e Resultados Obtidos

- **Conexão:**

- Verifica-se que a backdoor conecta-se com sucesso ao servidor.
- O servidor recebeu a conexão e iniciou a comunicação.

- **Execução de Comandos:**

- Comandos simples como `ls`, `pwd` e `cd` foram enviados e executados com sucesso na máquina alvo.
- A saída dos comandos foi enviada de volta ao servidor e exibida corretamente.

- **Upload e Download:**

- Foram realizados testes de upload e download na máquina-alvo com sucesso.

- **Keylogger:**

- Apesar de ser possível utilizar o keylogger, ele duplica as teclas que estão a ser recolhidas.

- **Screenshot:**

- Função que nos permite tirar um screenshot e guardá-lo na nossa máquina.

5 Análise de Segurança

5.1 Vulnerabilidades Exploradas pela Backdoor

A backdoor explora a falta de monitoramento e controle de conexões de saída na máquina alvo. Além disso, a ausência de mecanismos de autenticação robustos permite a execução remota de comandos.

5.2 Medidas de Detecção e Prevenção

Para mitigar o risco de backdoors, as seguintes medidas podem ser adotadas:

- Implementar firewalls para monitorar e bloquear conexões não autorizadas.
- Utilizar software de detecção de intrusão para identificar comportamentos anômalos.
- Aplicar atualizações de segurança regularmente para corrigir vulnerabilidades.

5.3 Melhorias na Segurança

- **Autenticação:**
 - Implementar autenticação de dois fatores para comandos remotos.
- **Criptografia:**
 - Utilizar criptografia para proteger a comunicação entre o servidor e a backdoor.

6 Demonstração do Funcionamento

Procedemos para a demonstração da nossa backdoor. Nesta demonstração iremos infectar um computador com o Kali Linux, enquanto que o servidor estará no Windows. Foi utilizada uma **Virtual Machine** conectada na mesma rede que está o computador que irá atacar.

6.1 Conexão

O primeiro passo consiste em criar uma conexão entre o servidor e o cliente por TCP/IP. O servidor (windows) irá conectar-se à porta 5555 e vai ficar à escuta. O cliente (Kali) irá correr o ficheiro python da backdoor e irá tentar conectar-se ao endereço ip do atacante na porta 5555.

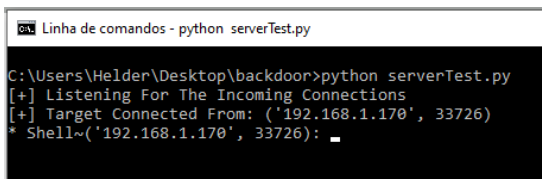


Figura 1: Conexão no Windows

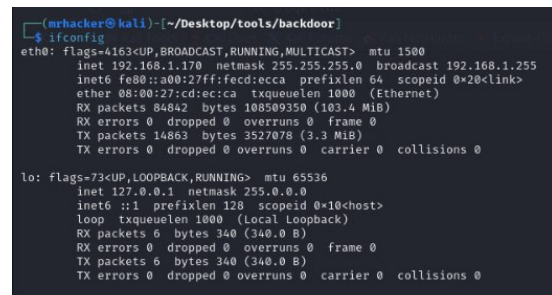


Figura 2: IP do kali

Como se pode observar nas figuras acima, depois de estabelecida a conexão, no terminal do Windows verificamos que conseguimos conectar ao IP 192.168.1.170, que corresponde ao IP do Kali Linux. Agora que temos acesso, vamos exprimentar alguns comandos.

6.2 whoami, pwd, dir, cd

Após estabelecer a conexão, o próximo passo é começar a explorar os diretórios e ver se conseguimos encontrar algo interessante. Como podemos ver na figura abaixo, conseguimos navegar no Kali com sucesso.

```

Linha de comandos - python serverTest.py
* Shell~('192.168.1.170', 33726): whoami
mrhacker

* Shell~('192.168.1.170', 33726): pwd
/home/mrhacker/Desktop/tools/backdoor

* Shell~('192.168.1.170', 33726): cd ..
* Shell~('192.168.1.170', 33726): pwd
/home/mrhacker/Desktop/tools

* Shell~('192.168.1.170', 33726): dir
backdoor webappentest

* Shell~('192.168.1.170', 33726): cd ..
* Shell~('192.168.1.170', 33726): pwd
/home/mrhacker/Desktop

* Shell~('192.168.1.170', 33726): dir
RatBackdoor.exe TheFatRat malware.py tester.py tools

```

Figura 3: Alguns simples comandos a serem testados

6.3 Upload e Download

Nesta secção vamos fazer upload e download de ficheiros na máquina infetada.

Upload do ficheiro:

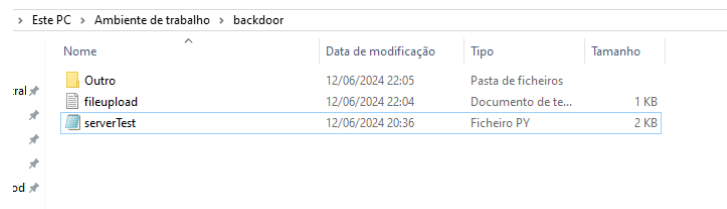


Figura 4: Pasta Windows com o ficheiro para upload

```

* Shell~('192.168.1.170', 33726): dir
RatBackdoor.exe TheFatRat malware.py tester.py tools

* Shell~('192.168.1.170', 33726): upload fileupload.txt
* Shell~('192.168.1.170', 33726): dir
RatBackdoor.exe TheFatRat fileupload.txt malware.py tester.py tools

```

Figura 5: Comando de upload

```

(mrhacker@kali)~[~]
$ cd Desktop

(mrhacker@kali)~/Desktop
$ ls
RatBackdoor.exe TheFatRat fileupload.txt malware.py tester.py tools

(mrhacker@kali)~/Desktop
$ cat fileupload.txt
0 ficheiro fez upload com sucesso

(mrhacker@kali)~/Desktop
$

```

Figura 6: Terminal do Kali

Download do ficheiro:

```
(mrhacker@kali)-[~]
$ cd Desktop

(mrhacker@kali)-[~/Desktop]
$ ls
RatBackdoor.exe TheFatRat filedownload.txt fileupload.txt malware.py tester.py tools

(mrhacker@kali)-[~/Desktop]
$ cat filedownload.txt
0 ficheiro foi descarregado com sucesso.

(mrhacker@kali)-[~/Desktop]
$
```

Figura 7: Terminal do Kali com ficheiro que irá ser descarregado para o atacante

```
* Shell~('192.168.1.170', 33726): dir
RatBackdoor.exe filedownload.txt malware.py tools
TheFatRat fileupload.txt tester.py

* Shell~('192.168.1.170', 33726): download filedownload.txt
* Shell~('192.168.1.170', 33726):
```

Figura 8: Comando de Download

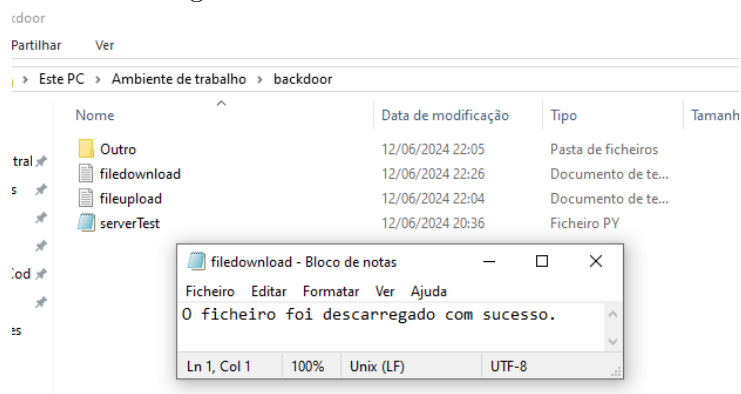


Figura 9: Pasta do Windows com o ficheiro descarregado

6.4 Keylogger

Relativamente ao keylogger, o funcionamento dele não é perfeito pois cada tecla premida é repetida 2x no output. No próximo exemplo vou chamar a função do keylogger e para testar vou escrever "Hello World".

```

Linha de comandos - python serverTest.py
* Shell~('192.168.1.170', 59520): keylog_start
* Shell~('192.168.1.170', 59520): keylog_dump
key.shift
key.shift
H'
H'
e'
e'
l'
l'
l'
l'
o'
o'
key.space
key.space
key.shift
key.shift
W'
W'
o'
o'
r'
r'
l'
l'
d'
d'
* Shell~('192.168.1.170', 59520):

```

Figura 10: Resultado do Keylogger

6.5 Screenshot

Por fim, temos uma função que tira um screenshot ao computador infetado e transfere a imagem para o diretório onde está a ser executado o servidor.

```

* Shell~('192.168.1.170', 60336): whoami
mrhacker
* Shell~('192.168.1.170', 60336): screenshot
[+] Screenshot saved as screenshot.png
* Shell~('192.168.1.170', 60336):

```

Figura 11: Terminal do windows com a execução da função screenshot

Este PC > Ambiente de trabalho > backdoor

Nome	Data de modificação	Tipo	Tamanho
serverTest2	12/06/2024 23:28	Ficheiro PY	3 KB
screenshot	12/06/2024 23:31	Ficheiro PNG	398 KB
fileupload	12/06/2024 22:04	Documento de te...	1 KB
filedownload	12/06/2024 22:26	Documento de te...	1 KB
Outro	12/06/2024 23:23	Pasta de ficheiros	

Figura 12: Pasta do Windows que contém o screenshot

7 Conclusão

Este trabalho demonstrou a implementação e funcionamento de uma backdoor, destacando as vulnerabilidades que ela explora e as consequências de sua utilização. A experiência prática reforçou a importância de medidas de segurança robustas para proteger sistemas contra acessos não autorizados. A compreensão de como as backdoors operam é essencial para desenvolver técnicas eficazes de detecção e prevenção, contribuindo para um ambiente informático mais seguro.

8 Referências

- Documentação do Python: <https://docs.python.org/3/>
- Conceitos de Segurança Informática: <https://owasp.org/>