ANALISI ED IMPLEMENTAZIONE DI UN KEYLOGGER
IN AMBIENTE UNIX/LINUX

(KEYLOGGER ANALYSIS AND IMPLEMENTATION IN A UNIX/LINUX BASED SYSTEM)

UNIVERSITÀ DEGLI STUDI DI NAPOLI
"PARTHENOPE"

Articolo a cura di/Article edited by

FIORENTINO MICHELE

ANALISI ED IMPLEMENTAZIONE DI UN KEYLOGGER IN AMBIENTE UNIX/LINUX

KEYLOGGER ANALYSIS AND IMPLEMENTATION IN A UNIX/LINUX BASED SYSTEM

ABSTRACT

Spesso quando si scarica un software, soprattutto se tale software non è stato ottenuto in modi esattamente leciti o se richiede particolari permessi, ci si chiede se il suo scopo in realtà sia quello di infettare la macchina ospite per fini malevoli (malware). In particolare, cosa accadrebbe se iniettasse un virus in grado di registare ogni singolo tasto che battiamo su tastiera? Si potrebbe rischiare di esporre qualsiasi tipo di dato sensibile. Software di questo tipo prendono il nome di **keylogger**. Comprendere il funzionamento e la possibile implementazione di un keylogger può permettere di sviluppare una maggiore consapevolezza nei loro confronti, ed è quello che si cerca di raggiungere in tale articolo.

ABSTRACT

When downloading software, especially if this software has not been obtained exactly in lawful ways or if it requires particular permissions, we could wonder if its purpose is actually to infect the host for malicious purposes (malware). In particular, what would happen if it could inject a virus capable of recording every single keystroke we type on the keyboard? You could risk exposing any kind of sensitive data. Software of this type are called keyloggers. Understanding the functioning and the possible implementation of a keylogger can allow you to develop greater awareness towards them, and this is what we try to achieve in this article.

PAROLE CHIAVE

keylogger, keylogging, malware, virus, linux, analisi di un keylogger, implementazione di un keylogger.

KEYWORDS

keylogger, keylogging, malware, virus, linux, keylogger analysys, keylogger implementation.

INTRODUZIONE

L'ambiente di sviluppo per il keylogger è ricaduto sugli ambienti UNIX/LINUX in quanto sono estemamente diffusi nell'ambiente lavorativo informatico, ed è stato utilizzato il linguaggio C per avere, almeno in teoria, una quanto più stretta possibile interazione con il sistema operativo. Si è sviluppato un keylogger che invia dati in modo remoto. Non avendo personalmente la possibilità di effettuare un test su rete Internet, ci si è limitati ad utilizzare 2 computer collegati sotto la stessa rete: un computer si comportava da server (colui che riceveva i dati dal keylogger), e l'altro da client (la vittima, che attraverso il keylogger inviava i dati al server).

INTRODUCTION

The devolpment environment for the keylogger was chosen to a UNIX/LINUX environment, as the as they are estimated to be widespread in the IT working environment, and the C language was used to have, at least in theory, as close as possible interaction with the Operating System.
It has been developed a keylogger that sends data remotely. Personally, not having the possibility to carry out a test on the Internet, I limited myself to using 2 computers connected under the same network: one computer acted as a server (the one who received the data from the keylogger), and the other as a client (the victim, who through the keylogger sent the data to the server).

IL KEYLOGGER

Il **keylogging** rappresenta l'azione di registrare i tasti battuti su una tastiera, solitamente in segreto, così che la persona che sta utilizzando il dispositivo non sia consapevole di essere monitorata. I dati dovranno essere poi recuperati dall'utilizzatore del programma di logging. Un programma di questo tipo prende il nome di **registratore di tasti** o **keylogger**.[1]
Un keylogger può essere iniettato nel computer ospite in due modi: **fisicamente**, ad esempio attraverso penne USB (keygrabber)[2], che inserite all'interno di un computer cercano di installare autonomamente il virus, oppure **ingannando l'utente** portandolo ad installare da sé il keylogger, pensando invece di star installando qualche altro programma.[3]
In base al tipo, un keylogger può operare a livello utente o a livello kernel.[4]

THE KEYLOGGER

Keylogging is the act of recording keystrokes, usually secretly, so that the person using the device is unaware that they are being monitored. The data must then be recovered by the user of the logging program. A program of this type is called a **keystroke logger** or **keylogger**.[1]
A keylogger can be injected into the host computer in two ways: **physically**, for example via USB sticks (keygrabber)[2], which inserted into a computer try to install the virus autonomously, or by **tricking the user** into installing from himself the keylogger, thinking instead that he is installing some other program.[3]
Depending on the type, a keylogger can operate at the user level or at the kernel level.[4]

STRATEGIE DI ESECUZIONE DEI KEYLOGGER

Un modo semplice per indurre la vittima ad installare ed eseguire un keylogger, è cercare di ingannarlo facendogli credere di star eseguendo tutt'altro: può essere un software fantoccio creato da 0, oppure un software effettivamente esistente open-source modificato in modo che contenga anche la porzione di codice infetta (ed è il caso preso in esame).

KEYLOGGER EXECUTION STRATEGIES

An easy way to get the victim to install and run a keylogger is to try to trick him into thinking he's running something else altogether: it can be puppet software created from scratch, or actually existing open-source software modified so that it also contains the infected portion of code (and this is the case we considerate).

IMPLEMENTAZIONE IDEALE DI UN KEYLOGGER DA REMOTO

Si è utilizzato il linguaggio C, facendo uso delle socket di Berkeley per creare le connessioni client/server e di librerie che permettessero di gestire gli input intercettati. Si vuole avere un accesso diretto alla tastiera (e non leggere da un canale come stdin), in quanto la nostra intenzione è che il keylogger venga eseguito in background e che intercetti ogni singolo tasto premuto, indipendentemente da dove l'utente stia scrivendo.
In Linux possiamo facilmente accedere agli input da tastiera leggendone il buffer. Infatti basta pensare che tutto in Linux è un file, e le periferiche non fanno eccezione. Tale buffer si trova in un percorso del tipo "/dev/ … /*-kbd". Il protocollo input utilizza una mappa di tipi e codici per esprimere in userspace il valore di input dei dispositivi[5]. Per convertire i valori in simboli dobbiamo far riferimento ad una tabella di conversione pubblicata dallo stesso Linus Torvalds su Github[6].
Il keylogger potrà mantenere determinate informazioni sul computer infetto, che potrebbe mandare al server per effettuare operazioni specifiche.
L'installazione deve essere quanto meno invasiva e allarmante possibile, ed una volta eseguito per la prima volta, vorremmo che il

IDEAL IMPLEMENTATION OF A REMOTE KEYLOGGER

The C language has been used, making use of Berkeley sockets to create client/server connections and libraries that allow to manage intercepted inputs. You want to have direct access to the keyboard (and not read from a channel like stdin), as our intention is for the keylogger to run in the background and intercept every single keystroke, no matter where the user is typing.
In Linux we can easily access keyboard input by reading its buffer. In fact, just think that everything in Linux is a file, and peripherals are no exception. This buffer is located in a path like "/dev/ ... /*-kbd". The input protocol uses a map of types and codes to express the input value of the devices in userspace [5]. To convert the values into symbols we have to refer to a conversion table published by Linus Torvalds himself on Github[6].
The keylogger will be able to keep certain information on the infected computer, which it could send to the server to perform specific operations. The installation must be as least invasive and alarming as possible, and once run for the first time, we would like the

keylogger rimanga sempre attivo in background, e che venga eseguito automaticamente ad ogni successivo avvio del computer.

keylogger to always remain active in the background, and to run automatically each time the computer is started.

SISTEMI DI PREVENZIONE CONTRO I KEYLOGGER

Non risulta difficile immaginare che tutti gli attuali sistemi operativi utilizzino delle misure di prevenzione per proteggersi da malware di questo tipo, ed ovviamente ciò vale anche per qualsiasi distribuzione Linux.
In linea generale, più è sensibile il file (o l'informazione) a cui vogliamo accedere, meno il SO operativo tenderà ad essere permissivo. Accedere al buffer di una tastiera, o eseguire un programma in automatico, sono operazioni viste con sospetto e richiedono permessi superiori.

PREVENTION SYSTEMS AGAINST KEYLOGGERS

It is not difficult to imagine that all current operating systems use preventive measures to protect themselves from this type of malware, and obviously this also applies to any Linux distribution.
In general, the more sensitive the file (or information) we want to access is, the less the operating OS will tend to be permissive. Accessing a keyboard buffer, or running a program automatically, are operations viewed with suspicion and require superior permissions.

IMPLEMENAZIONE PRATICA DI UN KEYLOGGER DA REMOTO

L'implementazione di un keylogger come quello descritto nei capitoli precedenti risulta piuttosto complicato, soprattutto considerando i sistemi di prevenzione adottati dai vari SO e considerando che il keylogger possa operare solo a livello utente. Il problema principale è che tali operazioni devono essere effettuate come super utente. Inoltre, non è stato possibile individuare un modo semplice per permettere l'esecuzione automatica del keylogger ad ogni avvio. Dunque il keylogger avrebbe effetto solo all'esecuzione del programma infetto, e solo se questo viene avviatto come superutente. È possibile comunque mantenere attivo il keylogger anche dopo la terminazione del programma principale.

PRACTICAL IMPLEMENTATION OF A REMOTE KEYLOGGER

The implementation of a keylogger like the one described in the previous chapters is rather complicated, above all considering the prevention systems adopted by the various OSes and considering that the keylogger can only operate at the user level. The main problem is that these operations must be performed as a super user. Also, it was not possible to find an easy way to allow the keylogger to run automatically on every boot. So the keylogger would only take effect when the infected program is run, and only if it is started as superuser. However, it is possible to keep the keylogger active even after the main program is terminated.

CONCLUSIONE

Dai test effettuati e dai problemi riscontrati durante il percorso, è parso evidente che lo sviluppo di un buon keylogger è meno scontato di quello che sembri, e richiede in ogni caso una certa disattenzione e superficialità da parte dell'utente. Tuttavia è assolutamente possibile, ed anche con un keylogger non molto complesso si possono provocare danni enormi. Per questo motivo è sempre consigliabile scaricare software solo dai siti ufficiali e di guardare con sospetto a qualsiasi applicazione richieda permessi speciali quando non dovrebbe averne bisogno.

CONCLUSION

From the tests carried out and the problems encountered along the way, it became evident that the development of a good keylogger is less obvious than it seems, and in any case requires a certain carelessness and superficiality on the part of the user. However it is absolutely possible, and even with a not very complex keylogger can cause enormous damage. For this reason, it is always advisable to download software from official websites only and to be suspicious of any application that requests special permissions when it shouldn't need them.

RINGRAZIAMENTI

Si ringrazia il prof. Castiglione Aniello per avermi guidato, fornendomi idee e consigli, durante il percorso culminato in questo articolo.

ACKNOWLEDGEMENTS

Thanks to prof. Castiglione Aniello for guiding me during the journey that culminated in this article, providing me ideas and advices.

RIFERIMENTI - REFERENCES

1. Nyang, DaeHun; Mohaisen, Aziz; Kang, Jeonil (2014-11-01). "Keylogging-Resistant Visual Authentication Protocols". *IEEE Transactions on Mobile Computing*. **13** (11): 2566-2579. doi:10.1109/TMC.2014.2307331. ISSN 1536-1233. S2CID 8161528.
2. https://www.ijcsmc.com/docs/papers/March2013/V2I3201322.pdf
3. https://www.mobiletekblog.it/2011/07/keylogger-usb-registrare-digitazione-tastiera/
4. https://www.csoonline.com/article/3326304/keyloggers-explained-how-attackers-record-computer-inputs.html
5. https://www.kernel.org/doc/Documentation/input/event-codes.txt
6. https://github.com/torvalds/linux/blob/master/include/uapi/linux/input-event-codes.h