# CVE-2022-30333

## Opis podatności

RARLAB UnRAR przed wersją 6.12 w systemach Linux i UNIX umożliwia przechodzenie przez katalogi w celu zapisu do plików podczas operacji wyodrębniania (czyli rozpakowywania), co zademonstrowano poprzez utworzenie pliku ~/.ssh/authorized\_keys. UWAGA: nie ma to wpływu na WinRAR i Android RAR.

Osoba atakująca może utworzyć pliki poza docelowym katalogiem ekstrakcji, gdy aplikacja lub użytkownik będący ofiarą wyodrębni niezaufane archiwum. Jeśli będą w stanie pisać w znanej lokalizacji, prawdopodobnie będą w stanie wykorzystać ją w sposób prowadzący do wykonania dowolnych poleceń w systemie.

**Base Score:** 7.5 HIGH

**Vector:**  CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N

**Affected patch**: wersje wcześniejsze niż 6.12.

## Algorytm działania

Złośliwie spreparowane archiwa RAR mogą zawierać dowiązania symboliczne. Osoba atakująca może wyodrębnić dowiązanie symboliczne wskazujące poza katalogiem wyodrębniania, a następnie wyłuskać je za pomocą drugiego pliku.

Zapobieganie atakom na łącza symboliczne okazuje się skomplikowane, ponieważ archiwa RAR można tworzyć i rozpakowywać zarówno w systemach Windows, jak i Unix, które różnią się znacznie pod względem obsługi ścieżek w systemie plików. Dowiązania symboliczne mogą być również względne i bezwzględne. Oto przykłady złośliwych dowiązań symbolicznych dla systemów plików Unix:

Ścieżka: /tmp/shell

Link symboliczny: ../../../tmp/shell

Aby zapobiec atakom na dowiązania symboliczne na systemy Unix, unrar zabrania wszelkich dowiązań symbolicznych ze ścieżką bezwzględną, sprawdzając, czy pierwszy znak jest ukośnikiem (/), jednak to można ominąć.

Po sprawdzeniu dowiązania symbolicznego jest ono normalizowane przez unrar. Archiwum RAR mogło zostać utworzone w systemie Windows lub Unix i że te systemy operacyjne w znacznym stopniu obsługują ścieżki plików.

Aby mieć pewność, że archiwum RAR utworzone w systemie Windows będzie można rozpakować w systemie Unix, ukośniki odwrotne (\) są konwertowane na ukośniki do przodu (/).

Załóżmy, że osoba atakująca utworzyła archiwum RAR zawierające dowiązanie symboliczne typu FSREDIR\_WINSYMLINK z celem ..\..\..\tmp/shell. Ponieważ archiwum jest rozpakowywane w systemie Unix, cel dowiązania symbolicznego jest uznawany za bezpieczny, ponieważ nie wykryto sekwencji ../.

Jednakże, ze względu na normalizację DosSlashToUnix(), ostatecznym celem dowiązania symbolicznego jest ../../../tmp/Shell. Wykorzystując to zachowanie, osoba atakująca może zapisać plik w dowolnym miejscu docelowego systemu plików.

Jak to można wykorzystać w celu ataku?

Na przykładzie ataku na serwis pocztowy Zimbra: wiadomości e-mail są skanowane antywirusem i antyspamem, aby przeskanować załączniki w mailu one powinny być rozarchiwowane. W tym celu korzystują z narzędzia strony trzeciej – w tym przypadku UnRar.

Aby uruchomić UnRar można użyć proctego polecenia „unrar x archive.rar /tmp/extract” . To polecenie rozpakuje wszystkie pliki z archiwum archive.rar do katalogu /tmp/extract. Aplikacja lub użytkownik wywołujący to polecenie oczekuje, że pliki zostaną zapisane wyłącznie w katalogu /tmp/extract. Oprogramowanie opiera się na tym założeniu, aby zapewnić bezpieczne usunięcie wszystkich plików po ich przetworzeniu. Ta sieć bezpieczeństwa jest implementowana przez unrar i jest domyślnie włączona.

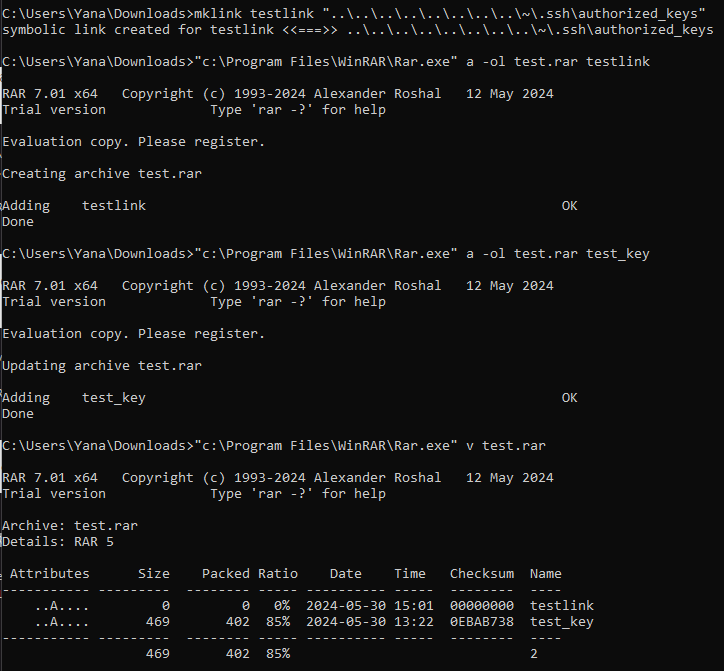
## Realizacja

Aby przetestować podatność stworzono archiwum zip zawierającym link symboliczny z drogą do /etc/ssh oraz wygenerowany klucz, aby póżniej nawiązać połączenie ssh za pomocą tego klucza. Archiwum rozpakowano na maszynie wirtualnej Linux za pomocą UnRar.

1. Utworzenie maszyny wirtualnej Linux do testowania i konfiguracja (instalowanie potrzebnej wersji UnRar)
2. Utworzenie archiwum zawierającego link symboliczny i plik z kluczem, który da możliwość utworzyć sesję ssh.
3. Połączenie do maszyny

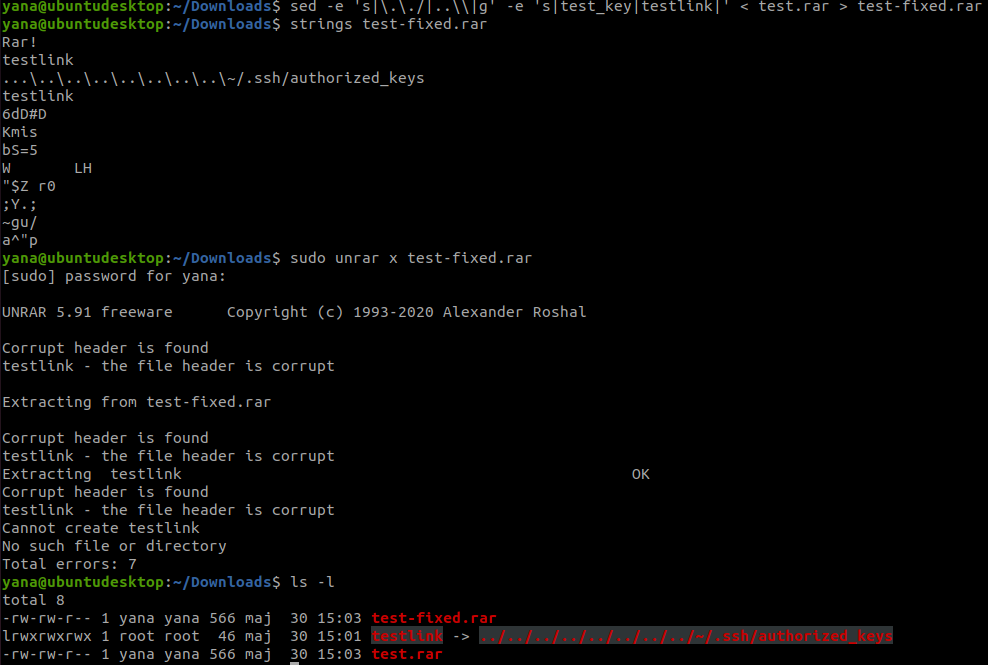
Testowanie przeprowadzone na maszynie wirtualnej Ubuntu. Zainstalowano wersję oprogramowania 5.9.

Aby wygenerować złośliwy plik .rar, używamy dowolnej wersji WinRAR (w systemie Windows) do kompresji łącza symbolicznego systemu Windows. Tworzymy dowiie symboliczny z celowym katalogiem /.ssh. Dodajemy ten link do archiwum. Następnie tworzymy drugi plik, którego nazwa jest tej samej długości co link, ale ma inną nazwę, a następnie dodajemy go do archiwum. W tym przypadku utworzono klucz publiczny ssh i nadano mu nazwę test\_key. Używane polecenia pokazano niżej:



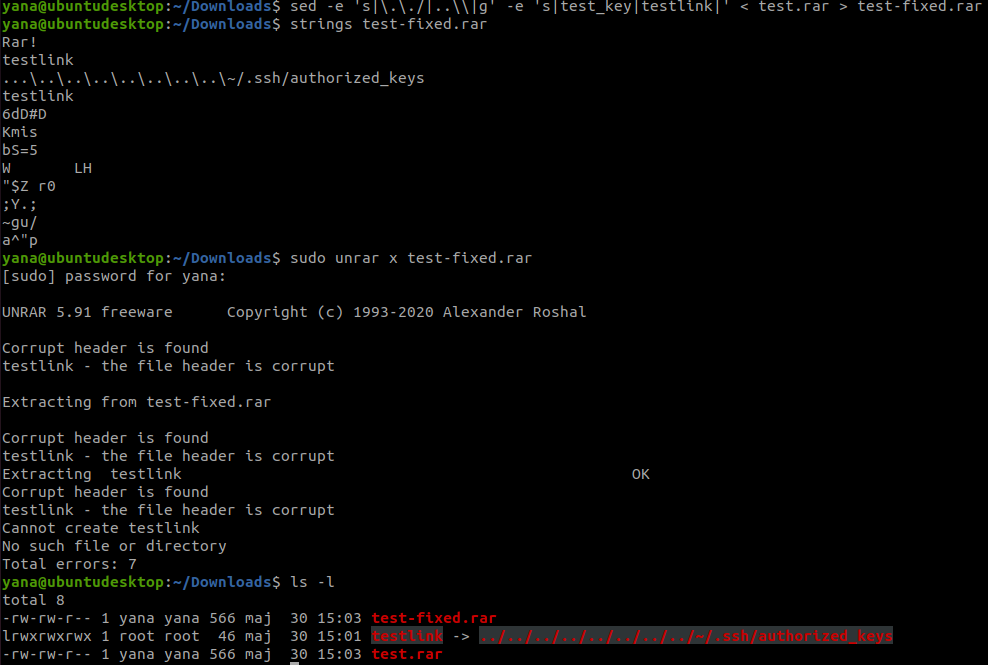
Rysunek 1. Tworzenie złośliwego archiwum

Następnie kopiujemy utworzony przez nas plik .rar na host Linux z podatną na ataki wersją unrar. Mimo że jest to Windows, WinRAR koduje ukośniki, więc zmieniamy je na ukośniki odwrotne za pomocą sed. Zmieniamy także nazwę pliku test\_key na testlink w archiwum. Naszym celem jest utworzenie dowiązania symbolicznego zwanego dowiązaniem testowym poprzez wyodrębnienie pierwszego pliku, a następnie zapisanie do dowiązania symbolicznego podczas wyodrębniania drugiego pliku. Oznacza to, że musimy przechowywać dwa pliki o tej samej nazwie, czego rar nam nie pozwala (jeśli spróbujemy, drugi plik nadpisze pierwszy), więc robimy to za pomocą sed.



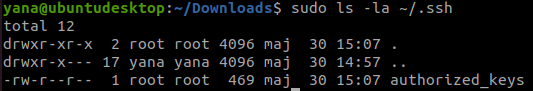
Rysunek 2. Edytowanie złośliwego archiwum

Następnie wyodrębniamy nasz złośliwy plik .rar za pomocą podatnej na ataki wersji unrar:



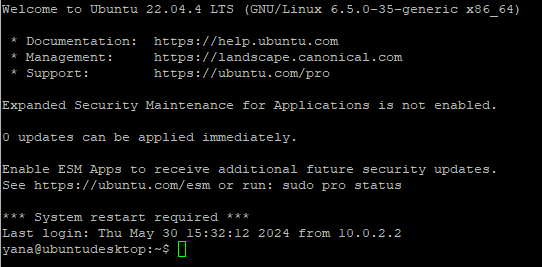
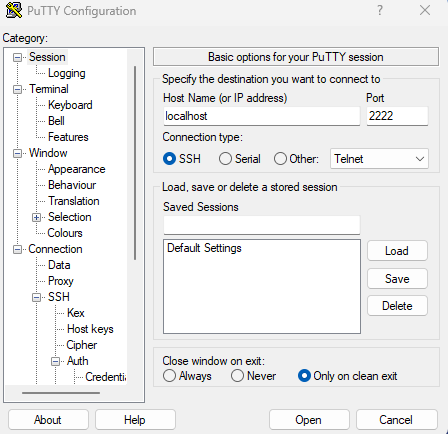
Rysunek 3. Rozpakowanie złośliwego archiwum

Mimo że wypisuje błędy, nadal pomyślnie wyodrębnia dowiązanie symboliczne, a dowiązanie rzeczywiście wskazuje poza strukturę katalogów. Moglibyśmy naprawić te błędy, ponownie obliczając sumy kontrolne pliku .rar, ale ponieważ unrar nie przejmuje się błędami, będzie to działać dobrze.



Rysunek 4. Wypakowany plik

Za pomocą PuTTy próbójemy nawiazać połączenia korzystając ze zgenerowanego klucza:



Rysunek 5. Połączenie za pomocą PuTTy

## Skutki

Przykładem wykorzystania podatności jest atak na serwis pocztowy Zimbra.

Zimbra to rozwiązanie e-mail gotowe do zastosowania w przedsiębiorstwach, z którego korzysta ponad 200 000 firm, instytucji rządowych i finansowych. Instancje Zimbra stały się niedawno celem kampanii jednodniowych ataków, prowadzonych prawdopodobnie przez podmiot państwowy, których celem były instytucje europejskiego rządu i media.

Fakt, że do kradzieży wiadomości e-mail z indywidualnych kont użytkowników wykorzystano lukę 0-day, pokazuje, jak cenne dla atakującego jest przejęte konto e-mail i jak katastrofalny wpływ takich luk na organizację. Tajne dokumenty mogą zostać skradzione, zresetowane hasła lub podszywać się pod członków organizacji w celu naruszenia większej liczby kont.

W przypadku Zimbry udane wykorzystanie daje atakującemu dostęp do każdej wiadomości e-mail wysłanej i otrzymanej na zaatakowanym serwerze e-mail. Mogą po cichu korzystać z funkcji logowania backdoorem i kraść dane uwierzytelniające użytkowników organizacji. Dzięki temu dostępowi prawdopodobnie będą mogli eskalować swój dostęp do jeszcze bardziej wrażliwych, wewnętrznych usług organizacji. Jedynym wymaganiem dla tego ataku jest zainstalowanie na serwerze unrar, co jest oczekiwane, ponieważ jest wymagane do skanowania archiwów RAR pod kątem wirusów i sprawdzania spamu.

## Podsumowanie

Przeprowadzenie ataku nie było trudnym. Jednak dla wykorzystania exploita atakujący musi zapewnić możliwość otrzymania złośliwego archiwum przez ofiarę i jego wypakowania. Na przykładzie Zimbry – można to zrobić, jeżeli serwis korzysta z usług UnRar (w którym był znaleziony exploit), infekując serwer. Aby atak skończył się sukcesem ofiara musi mieć zainstalowaną wersje UnRar poniżej 6.12. W wyniku atakujący może zapisać dowolne pliki w dowolnym miejscu na maszynie ofiary, co znaczy, że może otrzymać dostęp do maszyny, zainstalować inne złośliwe oprogramowanie itd.

Aby chronić przed takim atakiem najłatwiejszym jest zainstalowanie nowszej wersji oprogramowanie. Zajmuje to minutę i nie jest trudnym do zrobienia.

# Linki

https://attackerkb.com/topics/RCa4EIZdbZ/cve-2022-30333/rapid7-analysis?referrer=blog

https://github.com/debian-calibre/unrar-nonfree/tree/master

https://ioflood.com/blog/install-unrar-command-linux/