



Un sistema reputation-based in logica P2P per il blocco del malware a livello di Sistema Operativo



Relatore

Prof. Francesco Palmieri

Candidato

Giovanni De Costanzo

Introduzione

La diffusione dei dispositivi informatici e dei servizi online è in enorme crescita negli ultimi anni.

Entro il 2021 si prevede che:

Il 58% della popolazione sarà connesso alla Global Internet
(44% nel 2016)

Ogni persona avrà in media 3,5 dispositivi connessi alla rete
(2,3% nel 2016)

* Dati forniti da Cisco

Introduzione

Tale crescita non è accompagnata da un'adeguata consapevolezza riguardo al tema della **sicurezza informatica**.

Sono infatti numerosi i pericoli che si nascondono sulla rete, come:

- Frodi
- Furti digitali
- Compromissione della privacy



Malware

I **malware** sono software creati con l'intenzione di creare danni all'interno di un sistema.

Costituiscono uno dei principali pericoli per la sicurezza informatica.



Malware

In base all'azione intrapresa sono definite diverse categorie di malware:



Adware



Ransomware



Rootkit



Spyware



Trojan



Worm

Malware - casi noti

WannaCry è un ransomware creato nel 2017 e diffuso su sistemi Microsoft Windows sfruttando una vulnerabilità trapelata dalla NSA, nota come EthernalBlue.

Attacco hacker mondiale: virus "Wannacry" chiede il riscatto, ospedali britannici in tilt. "Usato codice Nsa"

la Repubblica.it

Oops, your files have been encrypted!

What Happened to My Computer?
Your important files are encrypted.
Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?
Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>. But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled. Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?
Payment is accepted in Bitcoin only. For more information, click <About bitcoin>. Please check the current price of Bitcoin and buy some bitcoins. For more information click <How to buy bitcoins>. And send the correct amount to the address specified in this window. After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am

Send \$600 worth of bitcoin to this address:

bitcoin

Un "ransomware" lanciato su centomila sistemi in 105 Paesi. Chiede soldi in Bitcoin, incognita l'origine. Pagamenti in corso, rischio truffa. Colpita anche l'Italia

Malware - casi noti

Stuxnet è un worm realizzato nel 2009 dagli Usa in collaborazione con il governo israeliano, con lo scopo di contrastare il programma nucleare iraniano.



The screenshot shows a news article from The New York Times. At the top right, there are links for "SUBSCRIBE NOW" and "SIGN UP FOR NEWSLETTERS". The main headline reads "*Obama Order Sped Up Wave of Cyberattacks Against Iran*". Below the headline, it says "By DAVID E. SANGER JUNE 1, 2012". To the right of the author's name are social media sharing icons for Facebook, Twitter, Email, and Print, along with a "360" link. The article begins with a paragraph about President Obama secretly ordering attacks on Iran's nuclear facilities. It then continues with a detailed explanation of the Stuxnet worm and its development by the United States and Israel.

MIDDLE EAST

Obama Order Sped Up Wave of Cyberattacks Against Iran

By DAVID E. SANGER JUNE 1, 2012

WASHINGTON — From his first months in office, President Obama secretly ordered increasingly sophisticated attacks on the computer systems that run Iran's main nuclear enrichment facilities, significantly expanding America's first sustained use of cyberweapons, according to participants in the program.

Mr. Obama decided to accelerate the attacks — begun in the Bush administration and code-named Olympic Games — even after an element of the program accidentally became public in the summer of 2010 because of a programming error that allowed it to escape Iran's Natanz plant and sent it around the world on the Internet. Computer security experts who began studying the worm, which had been developed by the United States and Israel, gave it a name: Stuxnet.

Meccanismi di protezione

Gli attuali meccanismi esistenti non sono in grado di proteggere i sistemi da malware non ancora conosciuti.

Utilizzano un approccio **blacklist**, definendo una lista di malware noti che non è consentito eseguire.



STOP

Obiettivo

Realizzare un meccanismo di protezione in grado di bloccare qualsiasi malware, evitando l'insorgere di nuove epidemie.

Soluzione proposta

Utilizzare un approccio **whitelist**, definendo cosa è possibile eseguire e bloccando tutto il resto.



Soluzione proposta

Utilizzare un approccio **whitelist**, definendo cosa è possibile eseguire e bloccando tutto il resto.

Problema:

Il numero spropositato di software da analizzare e da aggiungere in lista rende il compito molto gravoso.

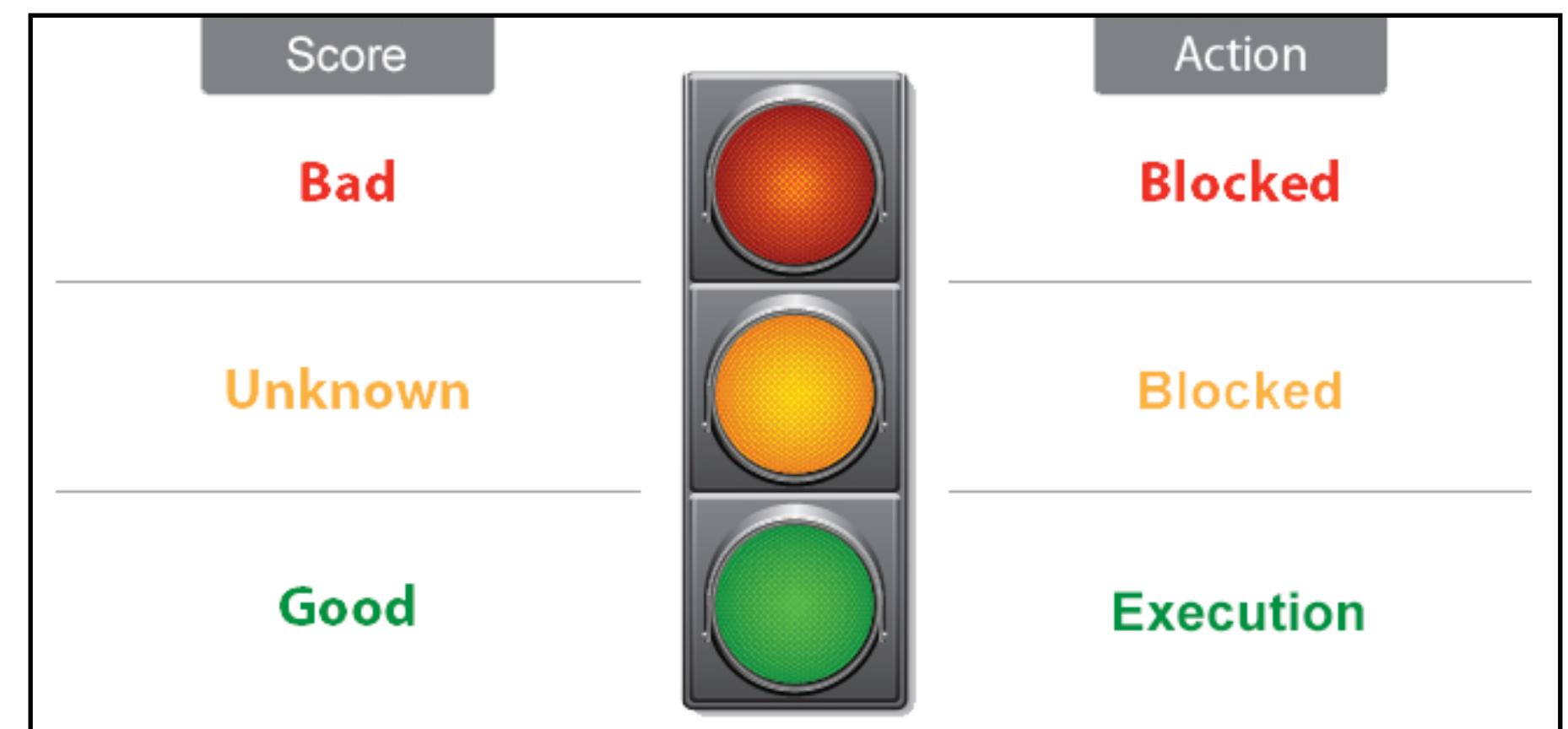
Idea:

Sfruttare la collaborazione degli utenti per realizzare un sistema di classificazione del software.

Classificazione del software

Il sistema di classificazione del software prevede che ad ogni file eseguibile sia associato un punteggio, determinato dalle segnalazioni degli utenti sulla rete.

Il punteggio è utilizzato dal sistema per classificare il software e stabilire se può essere eseguito.



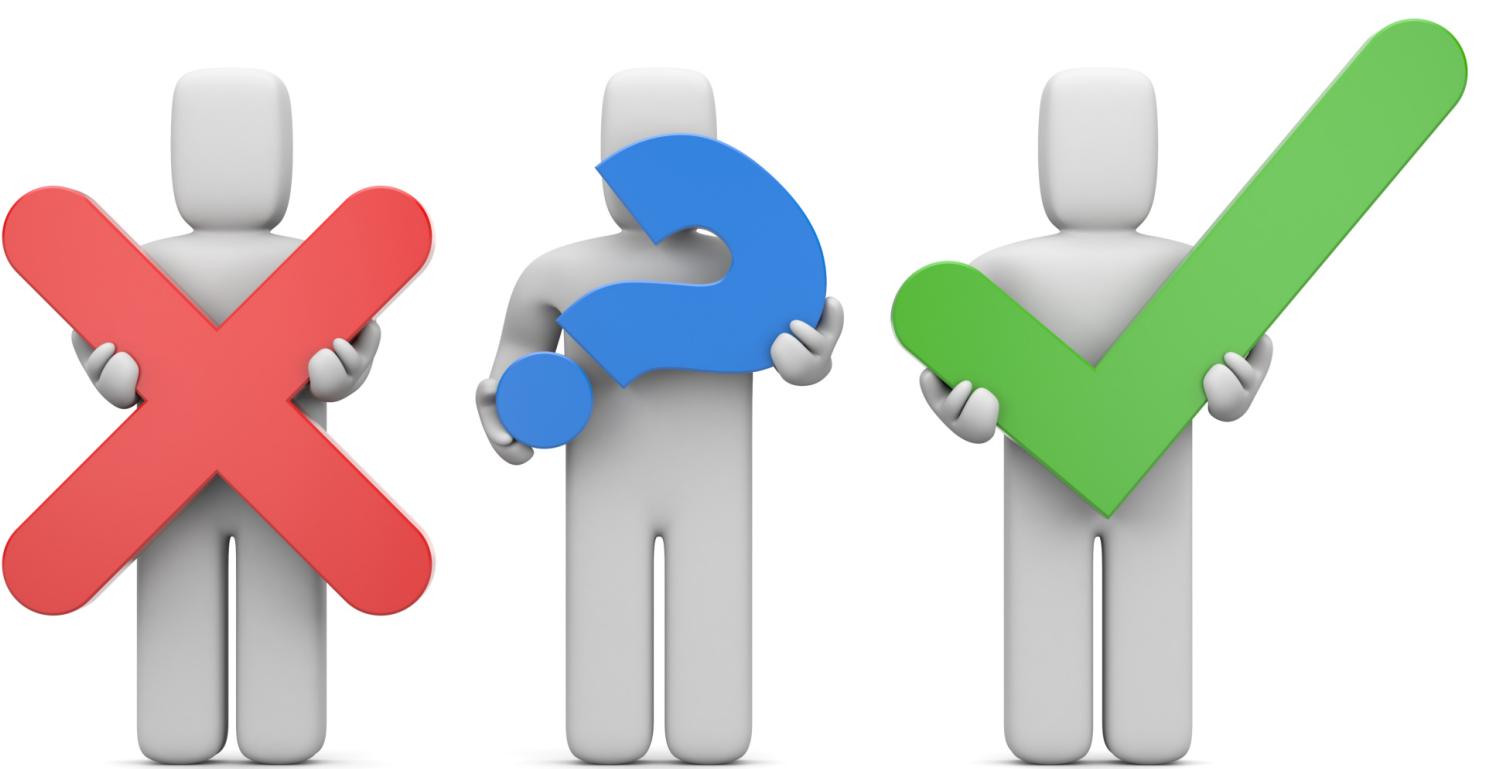
Reputazione degli utenti

Perché:

Per assicurarsi che il sistema di classificazione non venga sovvertito da utenti disonesti.

Funzionamento:

Ogni utente ha una reputazione determinata dal numero di segnalazioni corrette ed errate da lui inviate. La reputazione dà un peso alle segnalazioni che l'utente invia.



Descrizione del modello

- Per ogni file eseguibile è calcolato un **digest SHA-2** che lo identifica.



Descrizione del modello

- Per ogni file eseguibile è calcolato un **digest SHA-2** che lo identifica.
- Il software è classificato come *affidabile* o *malevolo* in base al valore di un **punteggio**.

Program	Score	
Notepad	1,1	✓
Keylogger	-0,3	✗
Calculator	1,0	✓
New_Program	0,5	✗

Descrizione del modello

- Per ogni file eseguibile è calcolato un **digest SHA-2** che lo identifica;
- Il software è classificato come *affidabile* o *malevolo* in base al valore di un **punteggio**;
- Il punteggio è determinato dalle **segnalazioni** inviate dagli utenti.



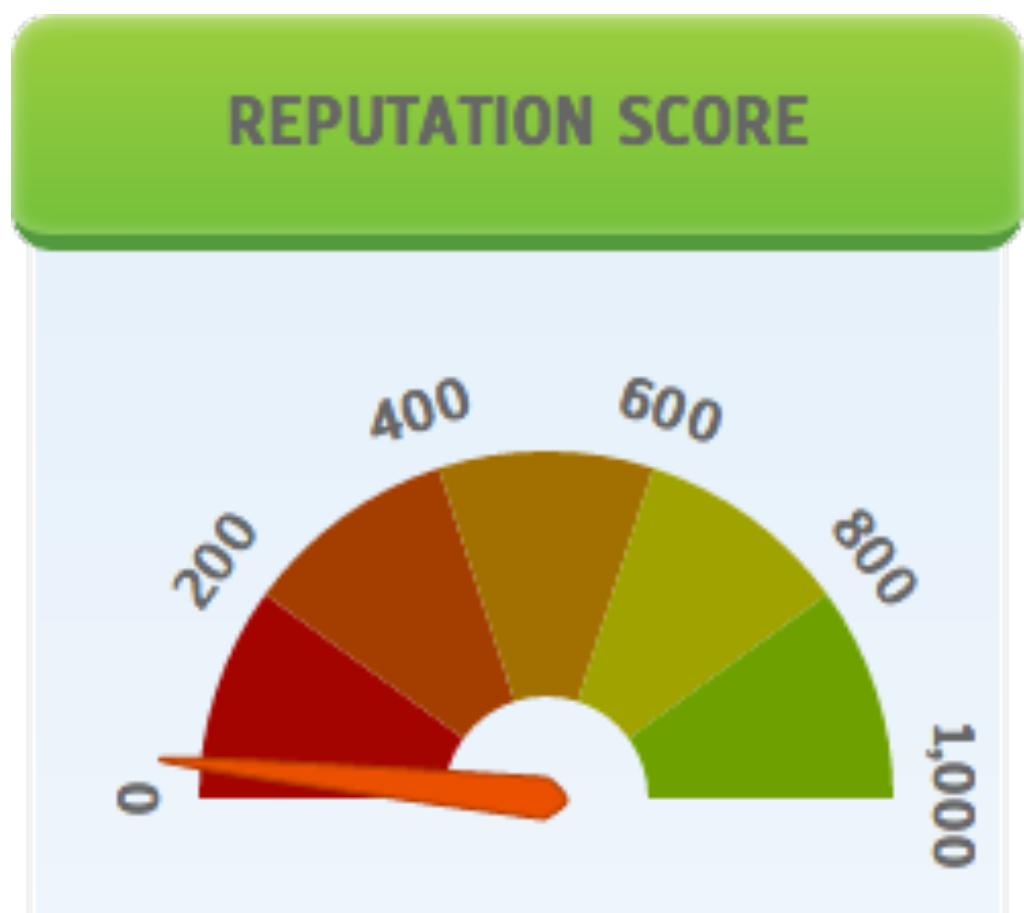
Descrizione del modello - [2]

- Ogni utente ha una **reputazione** che è determinata dal numero di segnalazioni corrette ed errate che ha inviato nel corso del tempo.



Descrizione del modello - [2]

- Ogni utente ha una **reputazione** che è determinata dal numero di segnalazioni corrette ed errate che ha inviato nel corso del tempo.
- Le segnalazioni di un utente sono pesate in base alla sua reputazione.



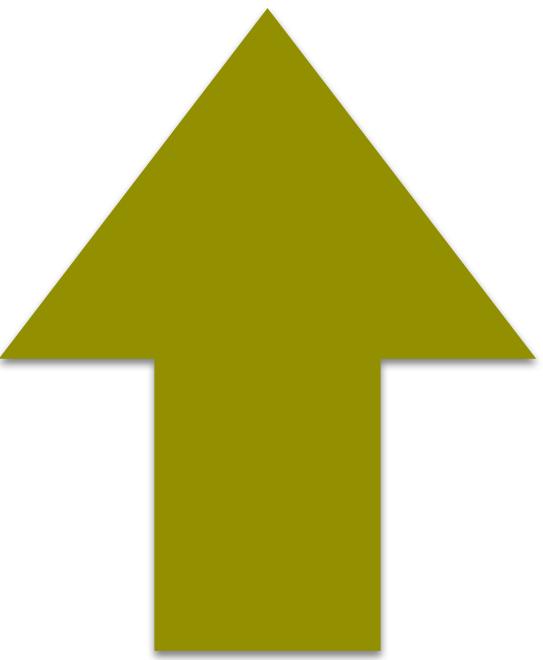
Descrizione del modello - [2]

- Ogni utente ha una **reputazione** che è determinata dal numero di segnalazioni corrette ed errate che ha inviato nel corso del tempo.
- Le segnalazioni di un utente sono pesate in base alla sua reputazione.
- Per stabilire se una segnalazione è corretta o meno, si attende che il punteggio relativo all'eseguibile raggiunga una soglia che ne consenta la classificazione.



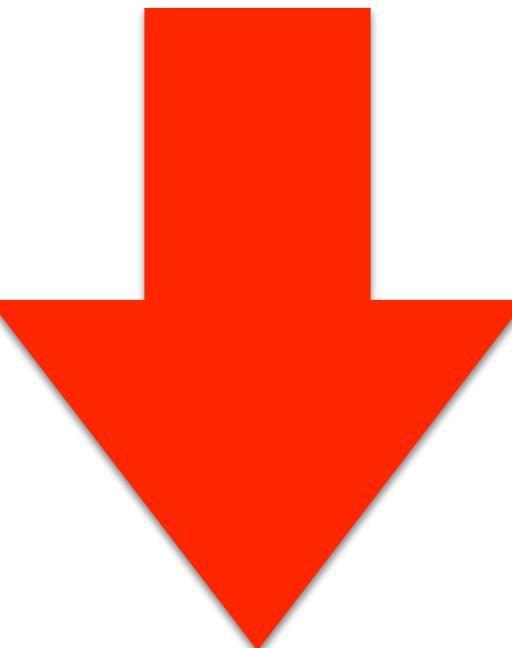
Descrizione del modello - (3)

- Per ogni **segnalazione corretta** un utente riceve un graduale **aumento** della propria reputazione.



Descrizione del modello - (3)

- Per ogni segnalazione corretta un utente riceve un graduale aumento della propria reputazione.
- Per una **segnalazione errata** invece l'utente riceve un considerevole **decremento** della reputazione.



Descrizione del modello - (3)

- Per ogni segnalazione corretta un utente riceve un graduale aumento della propria reputazione.
- Per una segnalazione errata invece l'utente riceve un considerevole decremento della reputazione.
- Il sistema raccoglie le segnalazioni e aggiorna il punteggio del relativo software alla chiusura di una **finestra temporale**.

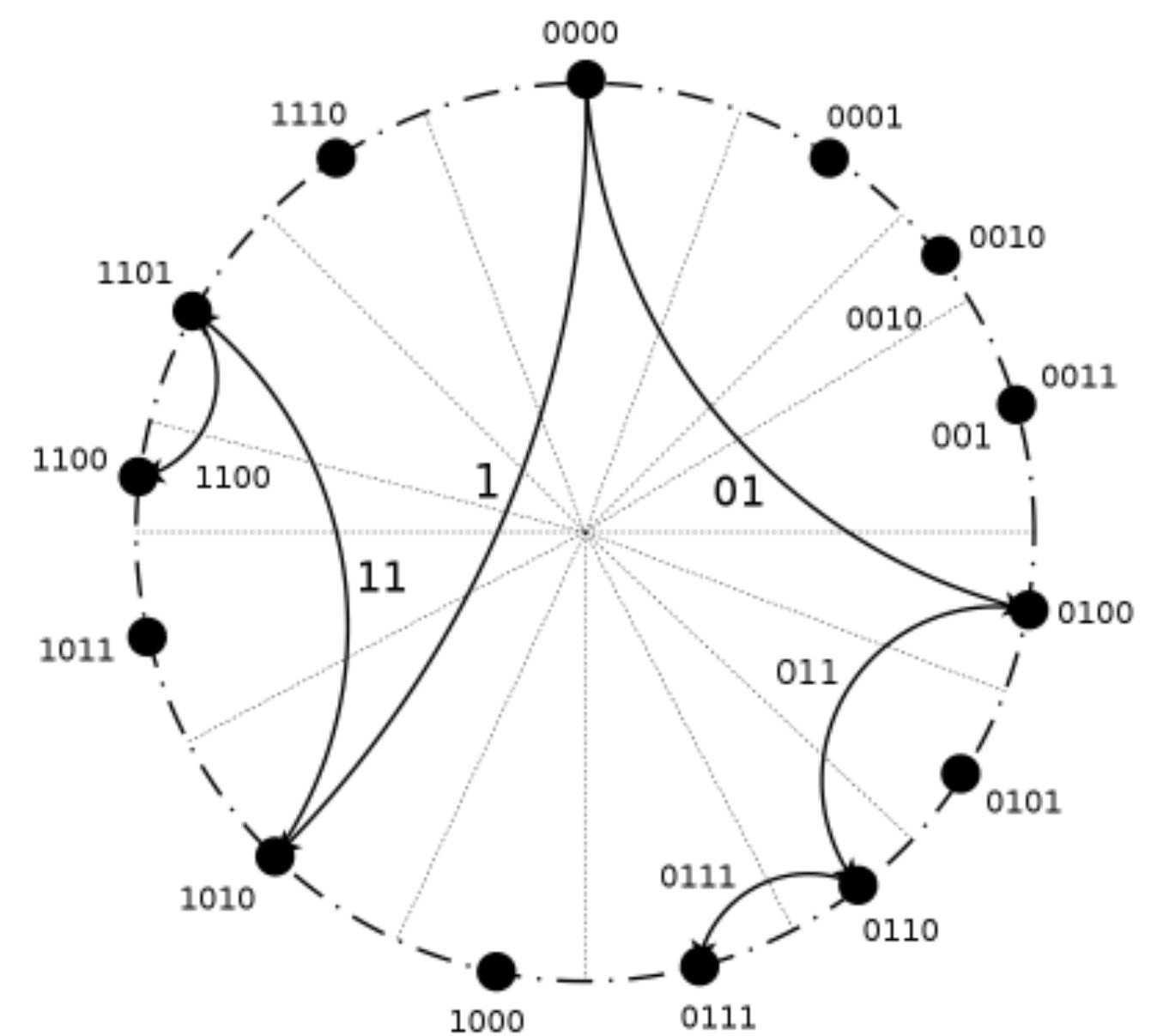


Kademlia DHT

Kademlia è un sistema distribuito che fornisce le funzionalità di una hash table per l'inserimento e il recupero di coppie chiave-valore.

Vantaggi:

- Scalabilità
- Tolleranza ai guasti



Kademlia DHT - [2]

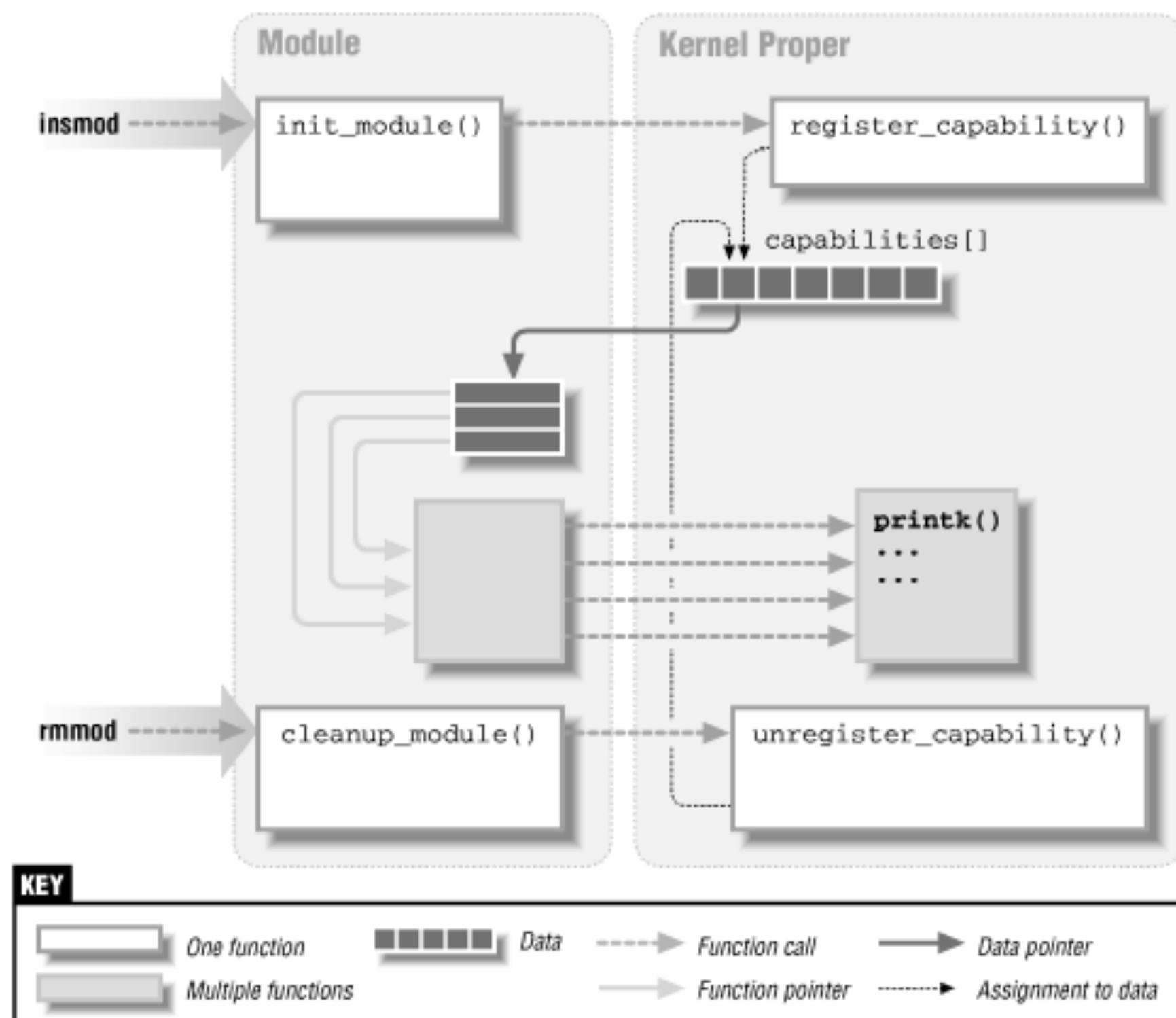
L'implementazione è stata modificata per consentire la **scrittura** sui nodi soltanto ad un server centrale, mediante firma digitale dei messaggi.



READ ONLY

Blocco del malware

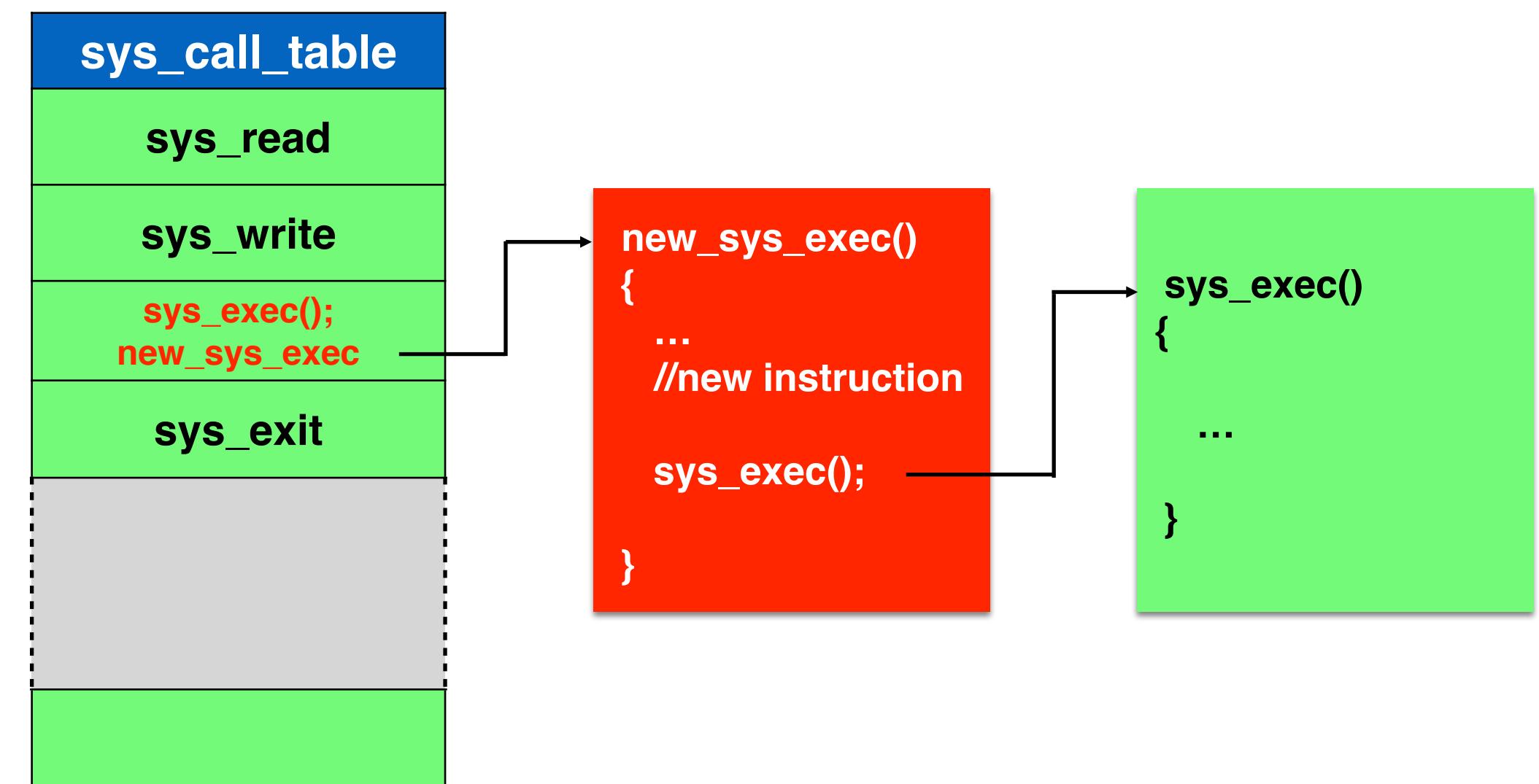
Il blocco del malware sul sistema operativo avviene mediante l'utilizzo di un **modulo kernel**.



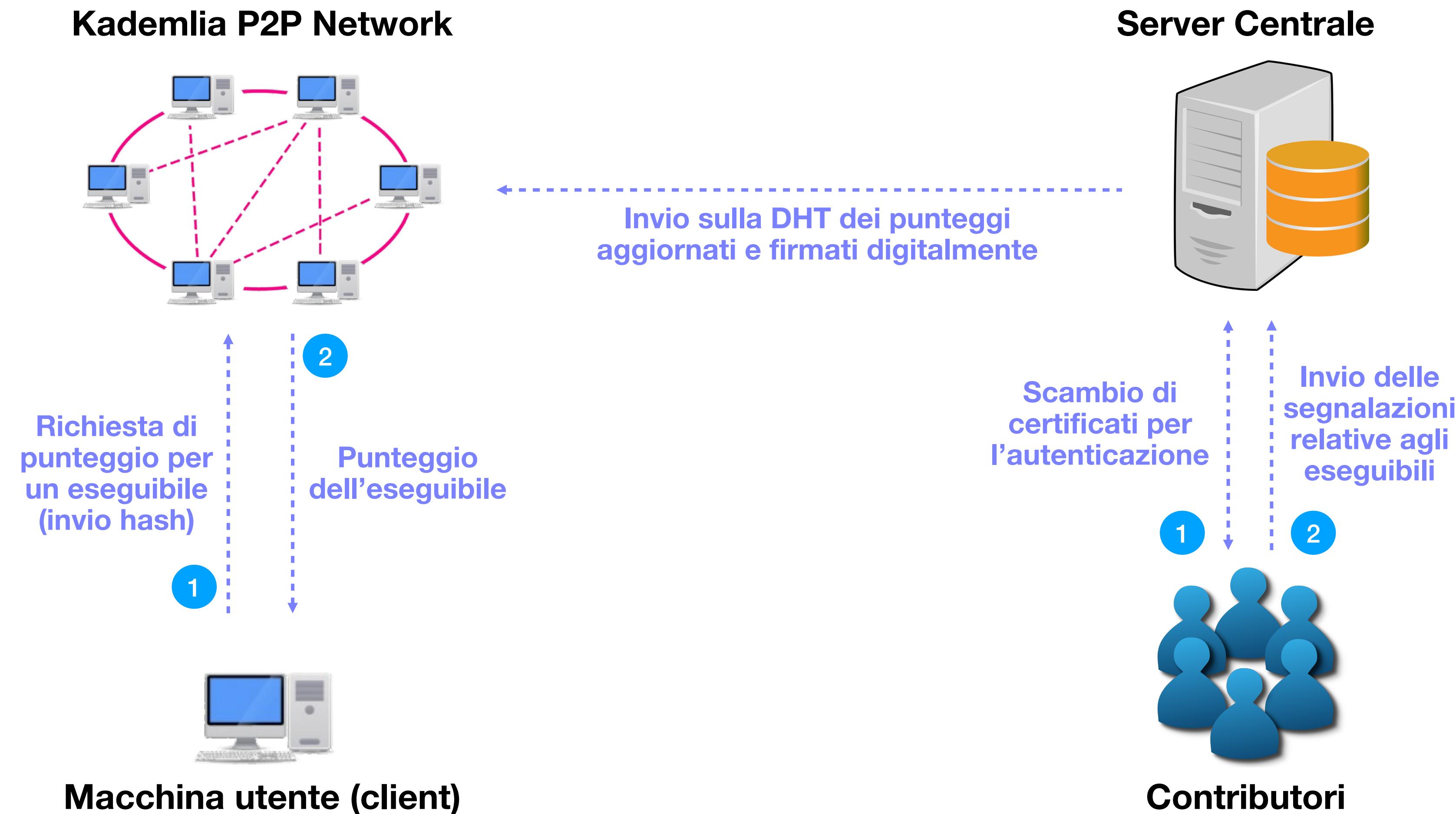
Blocco del malware - [2]

Viene attuata una tecnica di **hijacking** (dirottamento) per modificare il comportamento della chiamata di sistema execve.

Il modulo kernel intercetta le chiamate, e prima di procedere all'esecuzione di un file, controlla il suo punteggio sulla DHT usando l'hash come chiave.



Architettura del sistema



Overhead del modulo kernel

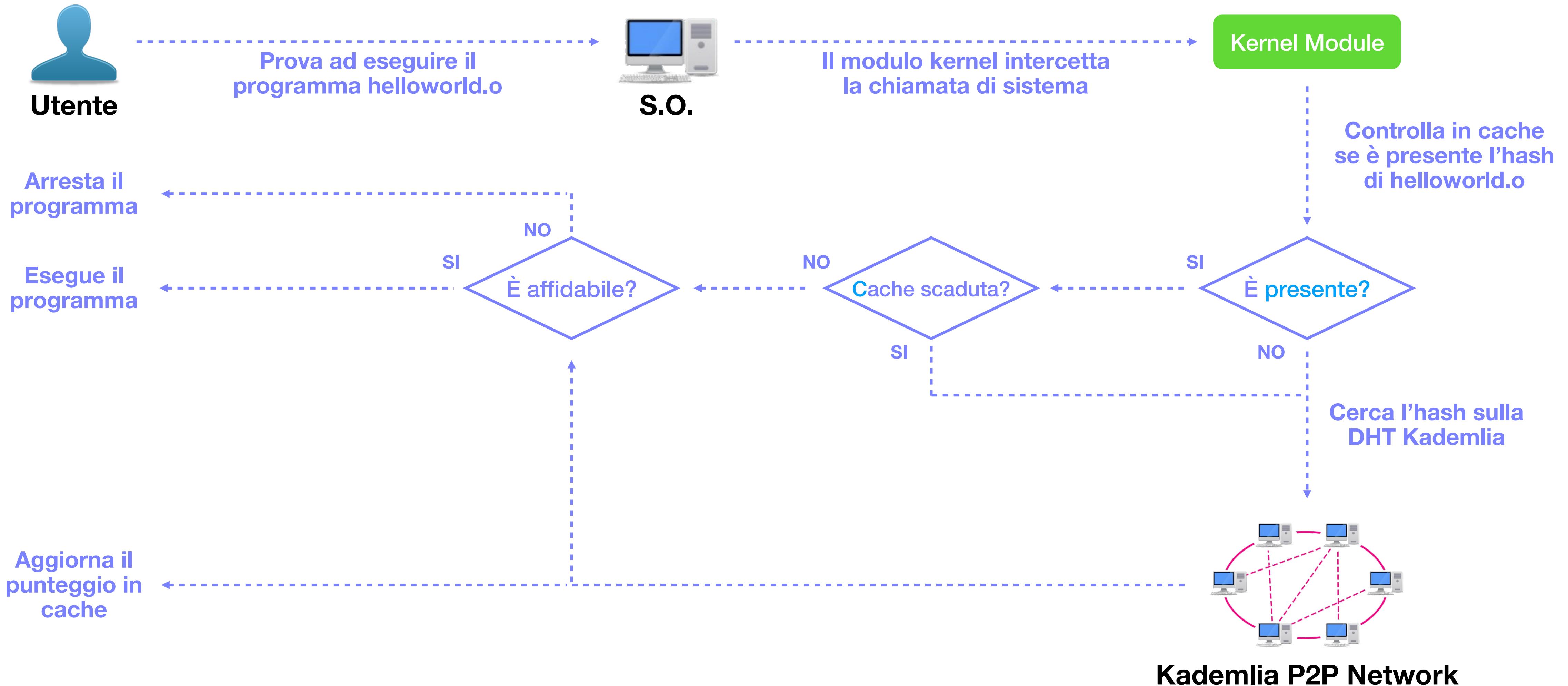
Problema:

Inviare una richiesta sulla rete prima di qualsiasi chiamata di esecuzione comporta l'introduzione di un overhead.

Soluzione:

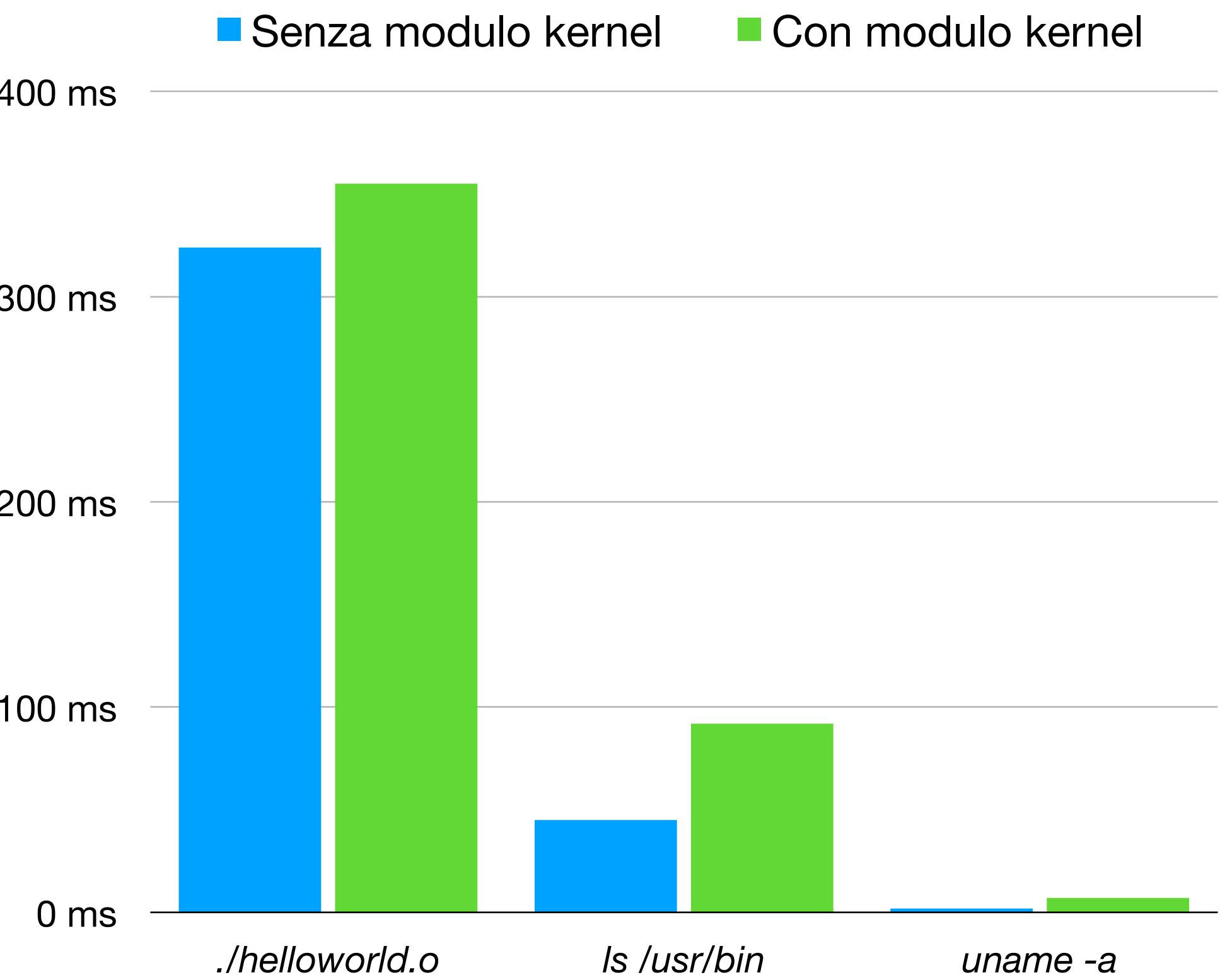
Per ridurre la latenza viene utilizzata una **cache** locale.

Schema di funzionamento



Prestazioni del sistema

Test effettuati in ambiente virtualizzato hanno mostrato una latenza media di 28 ms utilizzando la cache locale.



Limiti dell'implementazione

Quando si eseguono applicazioni scritte in linguaggi come Python oppure Java, l'eseguibile catturato dal modulo kernel è quello dell'interprete o della macchina virtuale.

Limiti dell'implementazione

Quando si eseguono applicazioni scritte in linguaggi come Python oppure Java, l'eseguibile catturato dal modulo kernel è quello dell'interprete o della macchina virtuale.

Possibili soluzioni:

1. Delegare il controllo dell'applicazione all'interprete o alla VM;
2. Modificare il modulo kernel affinché controlli gli altri parametri della chiamata, quindi il file dell'applicazione.

Conclusioni

Il sistema realizzato presenta un diverso approccio alla protezione contro il malware, basato su whitelist e quindi più conservativo.

Grazie all'utilizzo di una cache locale, la latenza introdotta alle chiamate di sistema è minima e non rende il sistema inutilizzabile.



Possibili sviluppi futuri

Rilevamento dei ransomware:

Analizzare l'entropia dei dati in scrittura effettuando l'hijacking della chiamata di sistema *write*.



GRAZIE PER L'ATTENZIONE