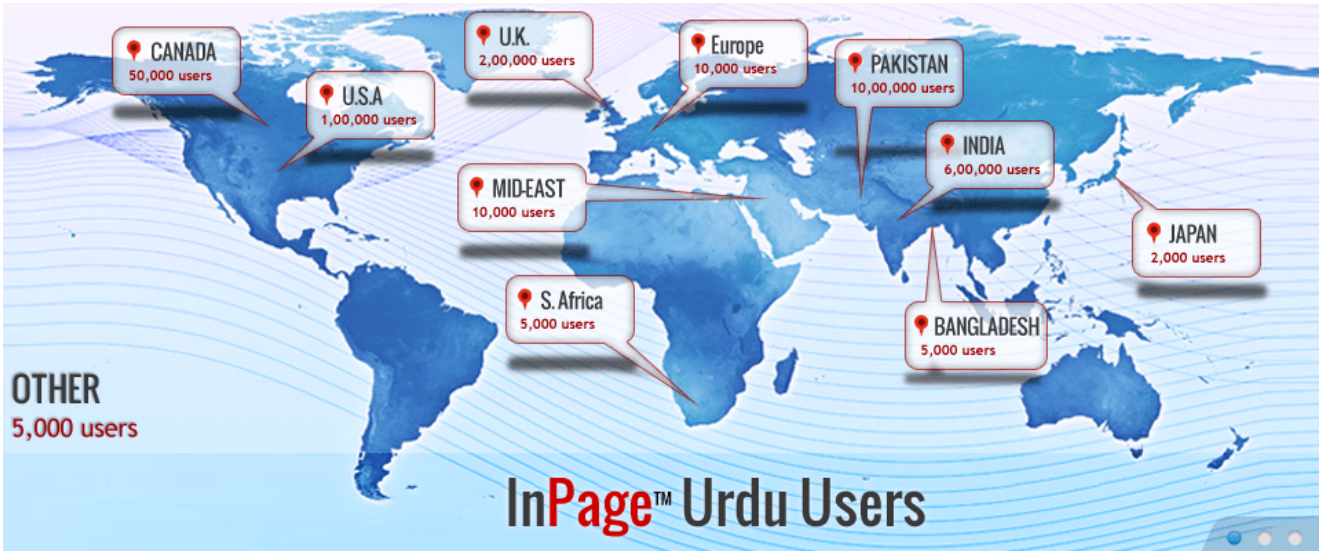
InPage zero-day exploit used to attack financial institutions in Asia

securelist.com/inpage-zero-day-exploit-used-to-attack-financial-institutions-in-asia/76717/

Authors

 Denis Legezo

In September 2016, while researching a new wave of attacks, we found an interesting target which appeared to constantly receive spearphishes, a practice we commonly describe as a “magnet of threats”. Among all the attacks received by this magnet of threats, which included various older Office exploits such as CVE-2012-0158, one of them attracted our attention. This file, which was also uploaded to a multiscanner service in September 2016, had an extension that we were unfamiliar with – “.inp”. Further investigation revealed this was an InPage document. InPage, in case you are wondering, is publishing and text processing software, mostly popular with Urdu and Arabic speaking users.



InPage user groups from vendor official site

Since no exploits for InPage have previously been mentioned in public, we took a closer look to see if the document was malicious or not. Further analysis indicated the file contained shellcode, which appeared to decrypt itself and further decrypt an EXE file embedded in the document. The shellcode appeared to trigger on several versions of InPage. We don't observe any public mentions of such exploit so we consider it a zero-day. We report it to the vendor and CERT-IN. Assigned vulnerability number is CVE-2017-12824.

Discovery and analysis

InPage is an interesting vulnerable software selection as it's widely used within the Indian Muslim population, as well as in Pakistan. This, of course, includes local mass-media and print shops, governmental and financial institutions (banks). If someone wants to deploy attack modules into regional press-related companies, an InPage exploit would work well.

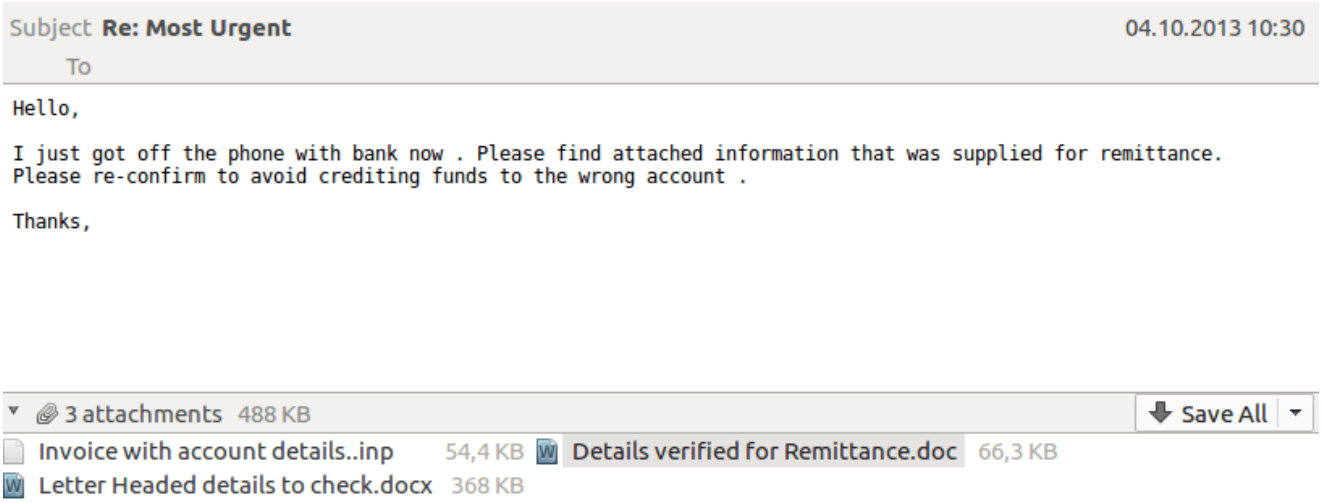
Due to its wide range of technologies, it wasn't perhaps surprising to see that Kaspersky Lab products already detect the exploit with the generic rule HEUR:Exploit.Win32.Generic. This detection is triggered by the presence of the shellcode inside a Microsoft Compound Storage file (OLE), which works extremely well for a wide category of Office-based exploits, going back to 2009.

The good news is that Kaspersky Lab users have been protected against this attack for quite some time – and the protection worked well in the past when it blocked a number of malicious InPage documents.

Between the various phishing campaigns relying on this exploit, one particular attack attracted our attention. The targets of this attack were special, since they were banks in Asia and Africa. The payload and C&C servers are also different from the recent attacks we've

observed, meaning there are probably several actors utilizing this zero-day exploit at the moment.

Technical details



Spearphishing e-mail with several malicious attachments. The .inp contains the zero-day exploit

In their attacks, the threat actors often use more than one malicious document. During spearphishing, the actors attached InPage files as well as .rtfs and .docs with old popular exploits.

Looking through all the related documents we could find, we counted several different versions of keyloggers and backdoors written mostly in Visual C++, Delphi and Visual Basic.

One such keylogger we analysed (MD5 hash: ***18a5194a4254cefe8644d191cb96da21***) was written in Visual C++. After gaining control, the module decodes several internal strings. One of them is the C2 domain name ***visitorzilla[.]com***. This backdoor maintains persistence by creating "**C:\Documents and Settings\<USER>\Start Menu\Programs\Startup\DataABackup.lnk**". Similar to the other campaign modules, it uses SetWindowsHook() with WH_KEYBOARD_LL hook to gather keystrokes. To gather keystroke data, the module uses two files on disk: C:\Documents and Settings\<USER>\Application Data\DataBackup\sed.ic and me.ic (located in the same directory).

Inside weaponized documents

InPage uses its own proprietary file format that is based on the Microsoft Compound File Format. The parser in the software's main module "inpage.exe" contains a vulnerability when parsing certain fields. By carefully setting such a field in the document, an attacker can control the instruction flow and achieve code execution.

The shellcode has three main parts:

1. Pattern searcher (so-called “egg hunter”) before the decoder,
2. Decoder.
3. Downloader.

The pattern searcher looks through all of the virtual memory space attempting to find the pattern “68726872”. Once the searcher identifies this pattern it starts the next stage of exploit – the decoder.

```
; ---------------------------------------------------------------------------
                fldpi                   ; entry point
                fstenv  byte ptr [esp-0Ch] ; save floating point environment
                pop     edi             ; get entry point address
                lea     ecx, [edi+19h]  ; get first encoded instruction address
                mov     edx, 2D4h       ; set encoded part length

decode:                                 ; CODE XREF: seg000:00001E21↓j
                not     byte ptr [ecx]  ; first not
                xor     byte ptr [ecx], 0ACh ; then xor
                inc     ecx
                dec     edx
                jnz     short decode
```

Shellcode decryptor

The small decoder obtains the instruction pointer and uses FLDPI + FSTENV instructions (an old and uncommon technique). The decoder is using an arithmetic NOT followed by a XOR 0xAC operation to decrypt the next stage.

Next, the downloader fetches a remote payload using InternetReadFile() and runs it using the WinExec() function in the %userprofile% directory. This functionality is very common and we’ve seen it with many other exploits. It’s the choice of vulnerable software that is interesting in this case and, for sure, the appearance of an exploit for software that is popular mostly in India and Pakistan.

The final payload is a Trojan written in Visual Basic 6. It defines a hook using the SetWindowsHook() function with the WH_MSGFILTER parameter. It communicates with its C2 server at 195.189.227.26 on port 8080.

During the initial session the C2 server sends “Pass” and host replies with “Auth<username>@<hostname>\#/<OS version>\#/<IP address>\#/-” In addition to b4invite[.]com this same Trojan was also spread using a configuration with the C2 server relaybg[.]com.

Victims

So far, victims of these attacks have been observed in Myanmar, Sri-Lanka and Uganda. The sector for the victims include both financial and governmental institutions.

Conclusions

By all appearances, this newly discovered exploit has been in the wild for several years. In some way, it reminds us of other similar exploits for Hangul Word Processor, another language/region-specific text processing suite used almost exclusively in South Korea. HWP has been plagued by several exploits in the past, which have been used by various threat groups to attack Korean interests.

Despite our attempts, we haven't been able to get in touch with the InPage developers. By comparison, the Hangul developers have been consistently patching vulnerabilities and publishing new variants that fix these problems. The best defense against exploits is always a multi-layered approach to security. Make sure you have an internet security suite capable of catching exploits generically, such as Kaspersky Internet Security. Installing the Microsoft EMET tool can also help, as well as running the most recent version of Windows (10). Finally, default deny policies, also known as allowlisting can mitigate many such attacks.

The Australian Signals Directorate Top35 list of mitigation strategies shows us that at least 85% of intrusions could have been mitigated by following the top four mitigation strategies together. These are: application allowlisting, updating applications, updating operating systems and restricting administrative privileges. Kaspersky Lab has technological solutions to cover the first three of these (i.e. all the technology-based strategies) as well as most of the others from Top35 ASD's list.

Kaspersky Lab detects this exploit as HEUR:Exploit.Win32.Generic.

More information about this exploit, associated campaigns and attacks is available to customers of Kaspersky Intelligence Services. Contact: intelreports@kaspersky.com

Indicators of compromise:

Hashes

f00e20ec50545106dc012b5f077954ae – rtf
729194d71ed65dd1fe9462c212c32159 – inp
c9e7ec899142477146d4f7f83df3f63f
750ed4f79496dee1d624a7b508f83f4e
B43aa5ea4ff5292fd92d416bb2b41c3a
4d508e44c5f3028a36a5206383cf235c
53c3503d3193bf14a93dc3ac24829490
5a9a8502b87ce1a6a608debd10761957

C&Cs used in the samples dropped by the weaponized InPage documents:

Relaybg[.]com
B4invite[.]com
Leastinfo[.]com
tropicmig[.]com
Digivx[.]com
Gigatrons[.]com
kinohata[.]ru
Visitorzilla[.]com
Ambicluster[.]com
Aliasway[.]com <- SINKHOLED by Kaspersky Lab
Xynoder[.]com
By4mode[.]com
Stringbit[.]com
Encrypzi.com
Gigsense[.]com
I3mode[.]com

- Spear phishing
- Vulnerabilities and exploits
- Zero-day vulnerabilities

Authors

 Denis Legezo

InPage zero-day exploit used to attack financial institutions in Asia

---

Your email address will not be published. Required fields are marked *