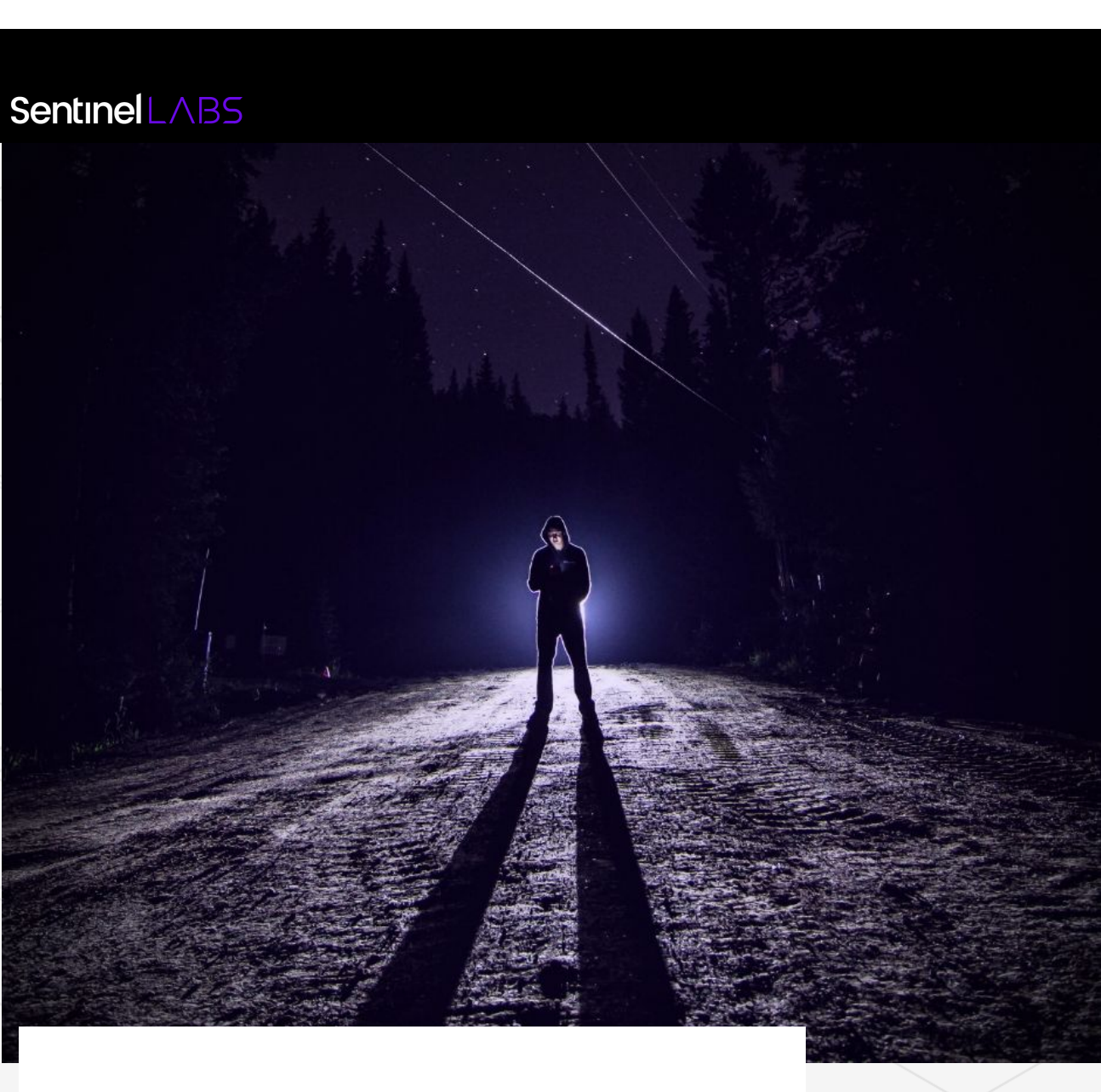
SentinelLABS



Vitali Kremez dissecting the 'Fin7' malware chain, which leverages malicious Office Macros and lightweight JS Loader scripts.

"FIN7" is a financially motivated advanced persistent group operating out of Eastern Europe. Since 2015, this group has continued to be extremely successful and formidable targeting various businesses seeking large-scale point-of-sale (PoS) compromises and network intrusion impacting global enterprises. The group is also known and notorious for its stealthy techniques and sophisticated and persistent approach.

Global corporations impacted by the group are primarily part of the restaurant, gaming, and hospitality industries. Some of the victims of this group include such restaurant chains as Chipotle Mexican Grill, Chili's, and Arby's.

Source: DOJ

Most interestingly, this group used a front company "Combi Security" (reportedly based in Russia and Israel) to recruit various hackers to join their activities. This front company allowed the group to sustain their hacking activities and truly professionalized their hacking approach.

Despite the previous arrests of three members of the FIN7 group in January 2018, the group and/or its remnants still remained active on the financial crime landscape.

Source: DOJ

It is notable that the group still deploys lightweight JavaScript backdoor with communication over HTTPS mimicking Content Delivery Network (CDN) domains.

Additionally, they still leverage JavaScript backdoor via renamed `wscript.exe` with the actual JavaScript code called "errors.txt," for example.

FIN7: From First-Stage Microsoft Office VBA Macro Loader to JS Loader

The FIN7 Microsoft document loaders do not rely on any exploits but simply require a social engineering trick to "Enable Content" to activate macros. Notably, to avoid process whitelisting of `wscript`, the macro logic copies the original JavaScript execution engine `wscript.exe` in `%LOCALAPPDATA%` and leverages a possible anti-analysis routine of checking the system drive size via `GetDrive.TotalSize` of more than 2456 bytes to possibly thwart anti-sandbox check.

The actual obfuscated Javascript backdoor is stored in UserForm object, which is also written to a disc as "errors.txt" in `%TEMP%`. The final execution of the backdoor is performed via this following command:

`%LOCALAPPDATA%mses.exe //b /e:jscript %temp%errors.txt`

Once it is done, the document macro runs a message box displaying "Decryption error" via `MsgBox("Decryption error")`.

Reversing Steps:

1. Extract the VBA macro via olevba;

2. Debug in Office VBA to retrieve decoded script;

3. Extract and prettify obfuscated JavaScript backdoor from userform object;

4. Modify JS code close to `eval()` and run script via Internet Explorer debugger, for example;

5. Debug, extract and beautify the full FIN7 JS backdoor.

FIN7 JS Loader/Backdoor XOR Encryption & Custom Encoding

The `crypt_controller` function accepts two parameters of *type* and *request*.

a. If *type* parameter equals "decrypt", the *request* is processed via `decodeURIComponent` splitting the request with separator `")*("` and then retrieving `encryption_key(second element[1])` from split request. If there's no encryption_key split, it pulls it as a random value via `(Math.floor(Math.random() * 9000) + 1000).toString().split("");`.

The decoding routine is a simple XOR loop decoding the content as follows joining the `result_string` via `.join` command.

```
var output = [];

    for (var i = 0; i < request.length; i++) {

        var charCode = request.charCodeAt(i) ^

encryption_key[i % encryption_key.length].charCodeAt(0);

        output.push(String.fromCharCode(charCode));
    }
```

b. If *type* parameter equals "encrypt", the `result_string` is joined with `")*("` and passed to `encodeURIComponent`.

FIN7 Second-Stage Machine & Network Profiling Script

In the aftermath of the initial call, the group deploys a custom "profiling" script meant to fingerprint the machine and the network environment more closely.

The malware checks for the presence of virtual machine, queries active directory, operating system, screen resolution, user account control (UAC) level, and retrieves a process list.

Finally, it formats the data and appends to "action=add_info" request, which is sent to the server.

Indicators of Compromise (IOCs):

Microsoft Office First-Stage VBA Macro ".doc" Documents:

SHA256:
`6e1230088a34678726102353c622445e1f8b8b8c9ce1f025d11bfffd5017ca82`
SHA256:
`f5f8ab9863dc12d04731b1932fc3609742de68252c706952f31894fc21746bb8`
SHA256:
`63ff5d9c9b33512f0d9f8d153c02065c637b7da48d2c0b6f7114deae6f6d88aa`

C2:

`googleapi-cdn[.]com`
`bing-cdn[.]com`
`cisco-cdn[.]com`

Recent Microsoft Office First-Stage VBA Macro ".xlsb" Documents:

SHA256:
`5fa5970548b43ae7d93d758a1eef1f12fd76891e36538e3ac170d5ab30906b5c`
SHA256:
`60dfe419dcba6dfe16d24f663b3393deeffdedbe4da468be63c63ec4b914d485`
SHA256:
`2ce1cfc137c0bcc82577cc77074c82154d81a7370491c85d43622af5186ef058`

Recent C2:

`realtek-cdn[.]com`

FIN7 | REVERSE ENGINEEERING | VITALI KREMEZ | ZERO2HERO

VITALI KREMEZ

Vitali Kremez is a strategic advisor for SentinelLabs. He specializes in researching and investigating complex cyberattacks, network intrusions, data breaches, and hacking incidents mainly emanating from the Eastern European cybercriminal ecosystem. He has earned the majority of major certifications available in information technology, information security, and digital forensics fields.