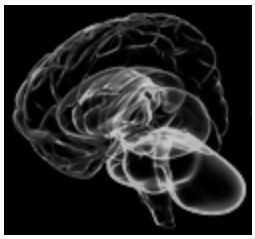
Zircolite vs Defense Evasion & Nobellium FoggyWeb

holisticinfosec.io/post/2021-09-28-zircolite/



I'm pleased to be back sharing outstanding tools for security practitioners with you after an extended time out to finish my Ph.D.
Here now, in our 145th installment of toolsmith, we discuss Zircolite, a standalone and fast SIGMA-based detection tool for EVTX or JSON, a fine tool brought to us courtesy of @waggabat. Zircolite's GitHub repo tells you absolutely everything you need to know, and the documentation is more than adequate, so I'll repeat only this:

- Zircolite is a standalone tool written in Python 3 allowing to use SIGMA rules on Windows event logs
- Zircolite can be used directly on the investigated endpoint or in your favorite forensic/detection lab
- Zircolite is fast and can parse large datasets in just seconds
- Zircolite can handle EVTX files and JSON files as long as they are in JSONL/NDJSON format
- Zircolite can be used directly in Python or you can use the binaries provided in releases

As you install Zircolite via a git clone to your preferred path, install the additional requirements via `pip install -r requirements.txt`.
Running Zircolite is as easy as the likes of `python zircolite.py --evtx logs --ruleset rules/rules_windows_generic.json`.
The rulesets provided are a set of SIGMA-based generic Windows rules and Sysmon rules. We owe Florian Roth and other many thanks for SIGMA, a true SOC work horse.
I'm particularly fond of actions we defenders can take to detect defense evasion. As such, our first test scenario is oriented accordingly. I created a *logs* directory in my Zircolite path and copied 16 defense evasion-specific EVTX log samples from Samir's killer Windows EVTX Samples repo. Love this resource! Thereafter, a quick run of Zircolite using the Sysmon rules provided immediate findings (two seconds) as seen in **Figure 1**.

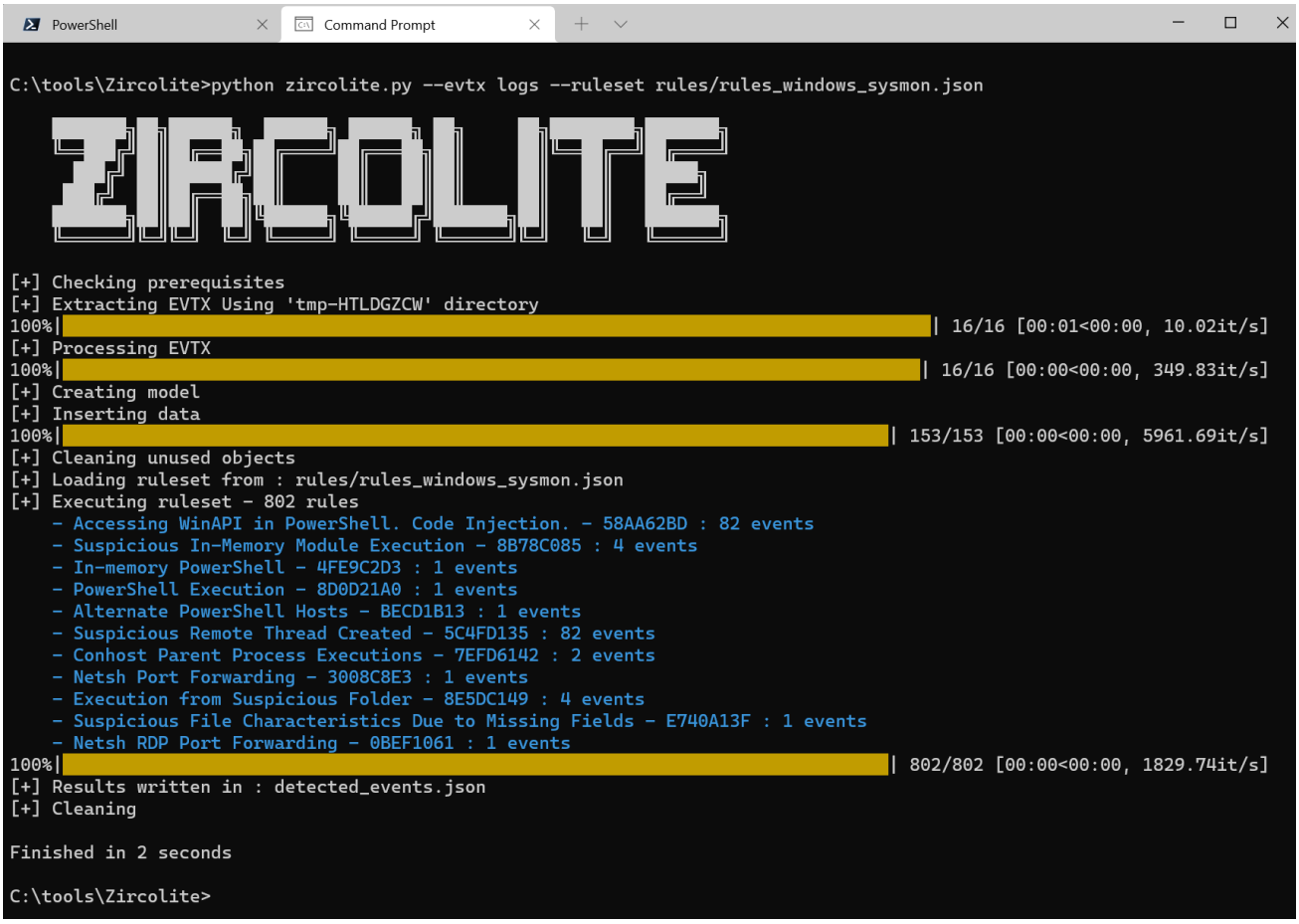


Figure 1: Zircolite run with Sysmon rules

Oh, wait. You're not running Sysmon everywhere possible? Tsk, tsk. Download Sysmon, and use SwiftOnSecurity's rocking good config file template with default high-quality event tracing. Thanks as always to Mark and Thomas for the indispensable masterpiece that is Sysmon. Zircolite findings are written out to *detected_events.json* in the Zircolite parent directory. In keeping with **Figure 2** represents the first of 82 detections of code injection with PowerShell.

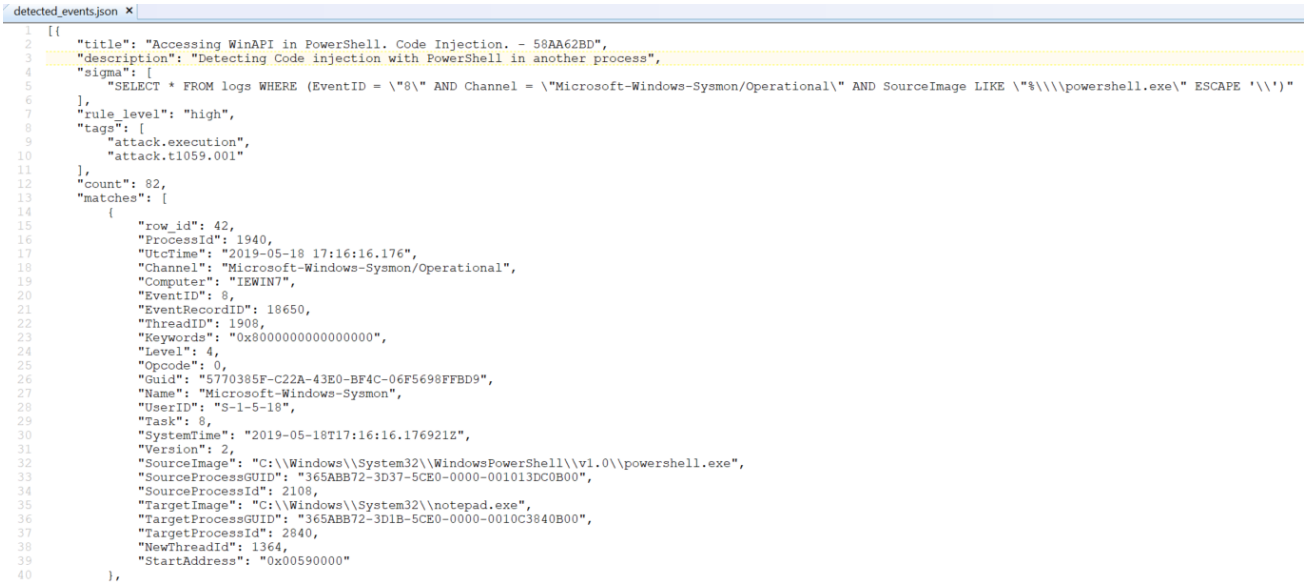


Figure 2: Zircolite detected event

Sweet, right?
Honing in on more specific adversarial actions per current events, first review the excellent deep dive analysis from Ramin, FoggyWeb: Targeted NOBELIUM malware leads to persistent backdoor. Herein, we learn about the latest NOBELIUM malware, a post-exploitation backdoor that Microsoft Threat Intelligence Center (MSTIC) refers to as FoggyWeb: “FoggyWeb is a passive and highly targeted backdoor capable of remotely exfiltrating sensitive information from a compromised AD FS server. It can also receive additional malicious components from a command-and-control (C2) server and execute them on the compromised server.” After compromising an AD FS server, NOBELIUM was observed dropping the following two files on the system:

%WinDir%\ADFS\version.dll
%WinDir%\SystemResources\Windows.Data.TimeZones\pris\Windows.Data.TimeZones.zh-PH.pri

This initial file drop struck me and others as a simple rule construct, I crafted a simple rule for use with Zircolite, resembling Florian’s, as a result.

To trigger the rule I simply created a small placeholder file with Textpad running with admin privileges and saved it to *C:\windows\ADFS*. This is not an actual image load event in the same sense as detected in Florian’s rule, just detection of the file creation in that path as captured by a Sysmon Event 11.

As I did, you can stick this JSON file in your rules directory or simply add it to the *rules_windows_sysmon.json* file. I left it standalone and ran `python zircolite.py --evtx logs --ruleset rules/rules_windows_nobellium_filedrop.json`. We have a hit, as seen in **Figure 3**.

Figure 3: Nobellium FoggyWeb file drop detected

As you can see, Zircolite can be adapted to almost any Windows-centric detection scenario as long as the events are written to EVTX. With Sysmon running Zirocite is an absolute no-brainer.

But wait! You need a GUI? No problem. Unzip the contents of *zircogui.zip* found in the gui directory. Revisting our Nobellium FoggyWeb scenario, I ran: `python zircolite.py --evtx logs/sysmon.evtx --ruleset rules/rules_windows_nobellium_filedrop.json --template templates/exportForZircoGui.tmpl --templateOutput gui/data.js`

The result, as seen in **Figure 4**, represents a convenient way to hunt about in numerous events per a MITRE ATT&CK category or an alert level.

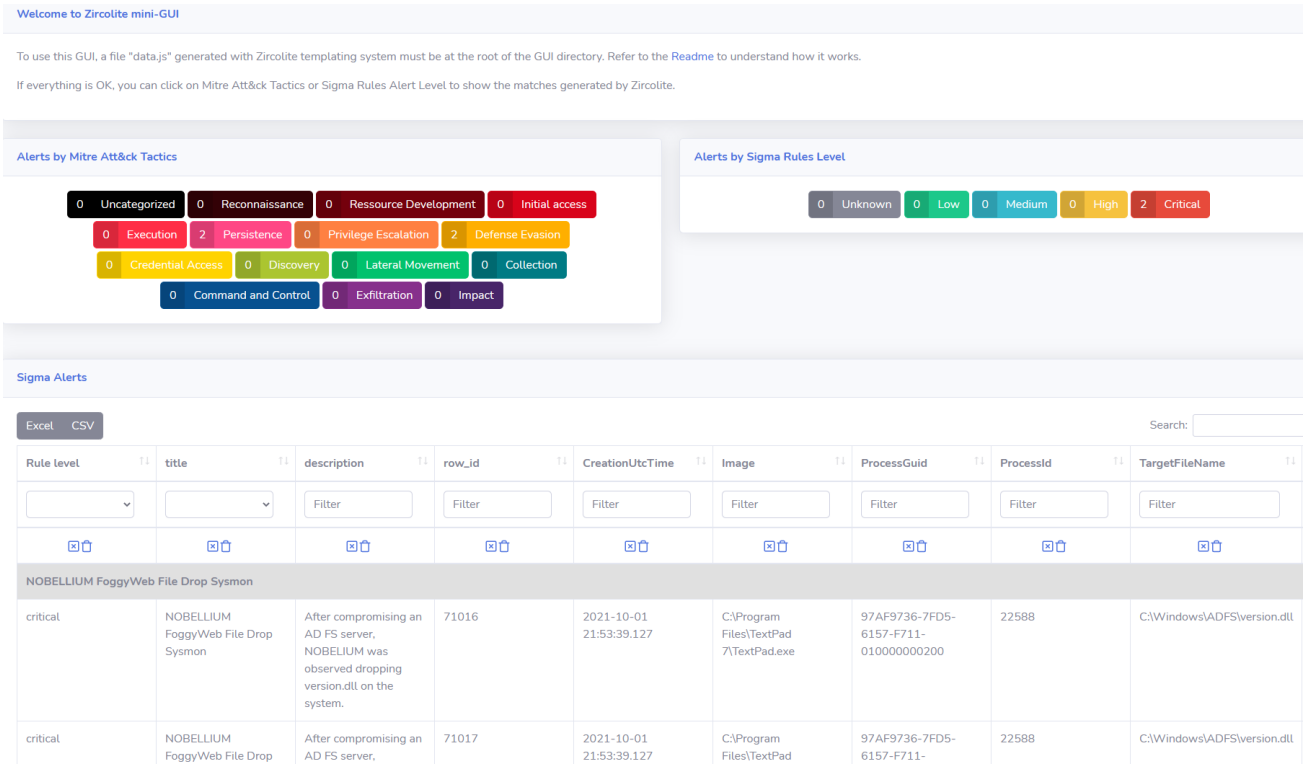


Figure 4: Zircolite GUI

I'm quite glad to be back at the keyboard working useful cybersecurity tooling scenarios with you. Zircolite represented a golden opportunity to do so and couldn't have coincided more nicely than with the recently released Nobellium FoggyWeb analysis. Great work from @waggabat, I hope they keep it up.

Cheers…until next time.

- ← Previous Post
- Next Post →