Building upon our previous post where we explored Velociraptor setup, we'll now delve into **investigating Potentially Unwanted Programs (PUPs)** using this powerful tool.

What are PUPs? Imagine programs that creep onto your system, often bundled with legitimate software, consuming resources and potentially compromising your privacy. While not full-blown malware, they're definitely unwelcome guests.

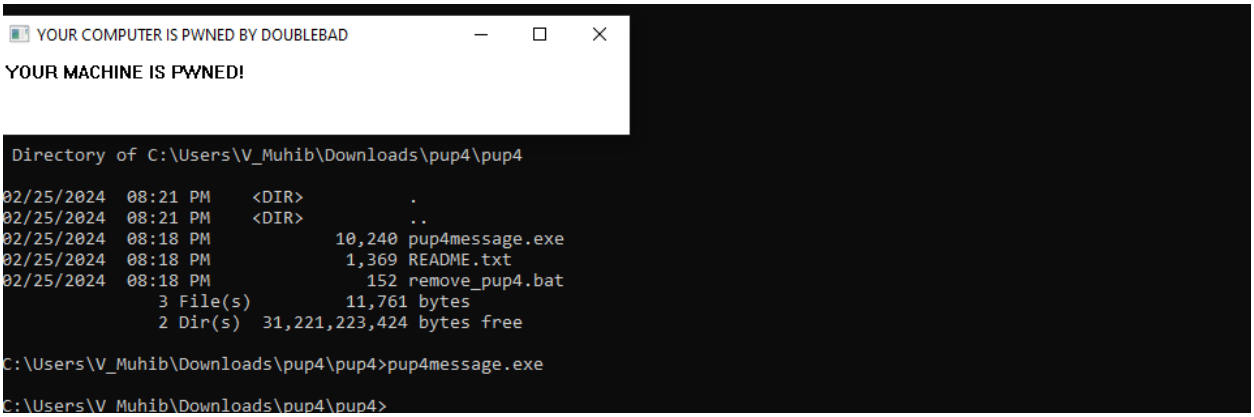
Since we do have;

1. A Debian Linux machine, running a Velociraptor server.
2. A Windows 10 machine, 64-bit, connected to the server as a Velociraptor client.

To infect the Windows machine with PUP, we'll download a simple malware and follow the following steps. Once the machine is infected, we'll use the Velociraptor GUI to gather data, analyze processes, and uncover potential threats. We'll use a VQL query to highlight unsigned binaries that can be potential security concerns due to their lack of verification.

Infecting the Windows Machine with PUP

1. Download this file: https://samsclass.info/152/proj/pup4.zip
2. Right-click pup4.zip. Click "Extract All..." Click Extract. Use the password "malware".
3. Run pup4\pup4\pup4.exe as administrator
4. Move pup4\pup4\pup4message.exe to C:\
5. Delete pup4.zip and the pup4 folder. Empty the Recycle Bin.
6. Restart your machine. If an "Open File - Security Warning" box pops up, uncheck the "Always ask before opening this file" box and click **Run**. 7. An irritating message pops up, as shown below.



Investigating with Velociraptor:

Access the Velociraptor GUI: Click the down-arrow next to the search box and select "Show All." Choose your client's ID.

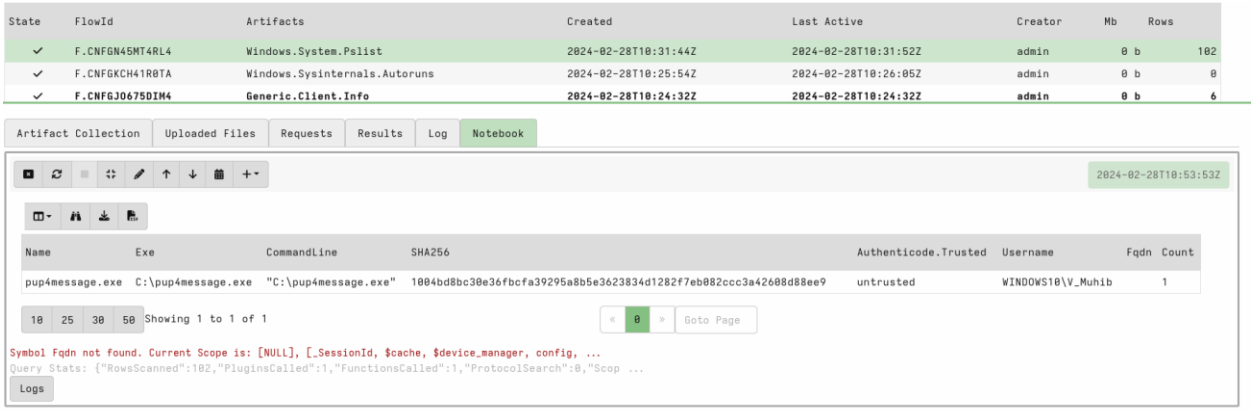
Gather data: Click "Collected" followed by the "+" sign and search for the "**Windows.System.Pslist**" collector. Launch this collector.

Analyze processes: Click "Results" to view the MD5 hash of each running process.

Uncover potential threats: Click "Notebook" and "Show All/Add Columns." Edit a cell and paste the following VQL query:

```
SELECT Name,Exe,CommandLine,Hash.SHA256 AS SHA256, Authenticode.Trusted AS Authentication, Username,
Fqdn, count() AS Count FROM source()
WHERE Authenticode.Trusted = "untrusted" // unsigned binaries
GROUP BY Exe

// Sort results ascending
ORDER BY Count
```

This query highlights unsigned binaries (lacking a digital signature) which can be potential security concerns due to their lack of verification.

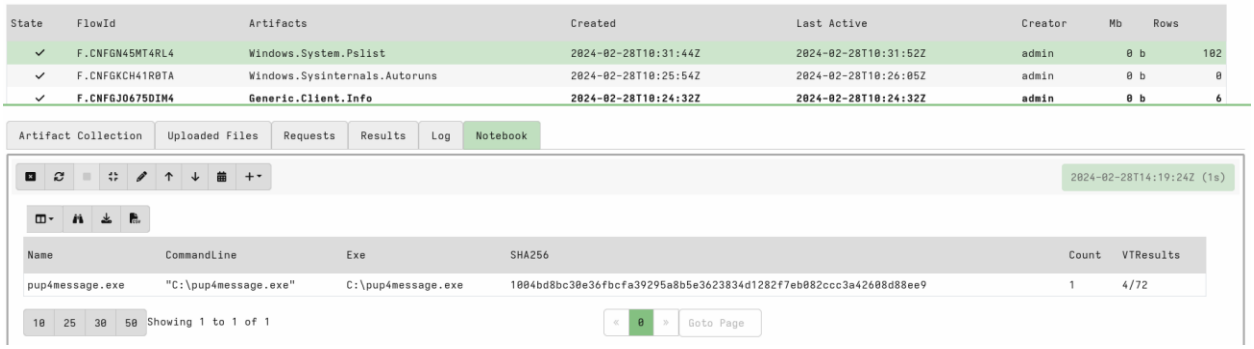


NOTE: Unsigned binaries are executable files (such as **.exe**, **.dll**, or **.so** files) that do not have a digital signature.

Further investigation (optional):
Virus Total: Obtain a free API key and paste this next query into the Notebook:

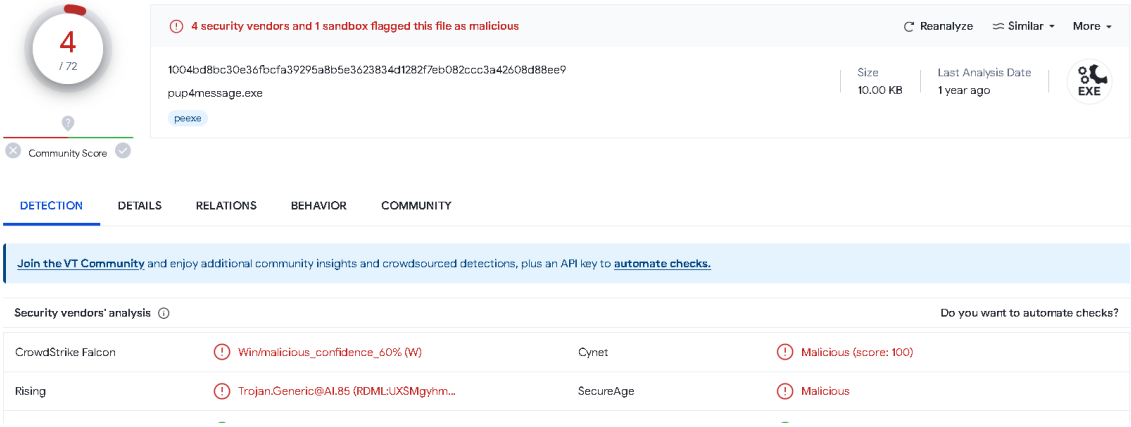
```
// Get a free VT api key
LET VTKey <= "YOUR API KEY"

// Build the list of untrusted processes first
Let Results = SELECT Name,CommandLine,Exe,Hash.SHA256 AS SHA256, count() AS Count FROM
source()
WHERE Authenticode.Trusted = "untrusted"
AND SHA256 // only entries with the required SHA256

// List of environment-specific processes to exclude
AND NOT Exe = "C:\\user-automation\\user.exe"
GROUP BY Exe,SHA256

// Now combine the previous query with the Server Enrichment query
SELECT *, {SELECT VTRating FROM Artifact.Server.Enrichment.Virustotal(VirustotalKey=VTKey,
Hash=SHA256) } AS VTResults FROM foreach(row=Results) WHERE Count < 10
```

ORDER BY VTResults DESC

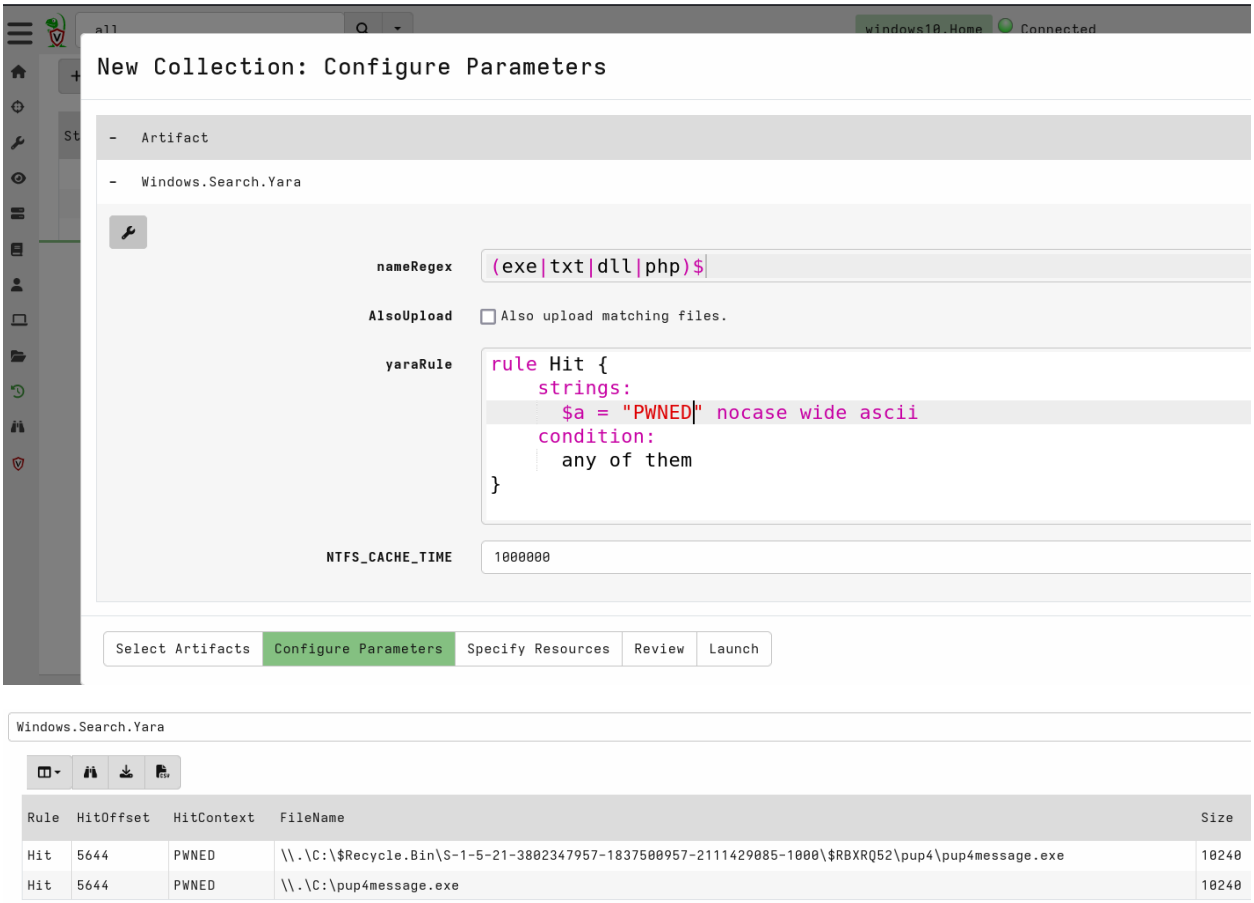


This checks the identified unsigned binaries against VirusTotal for potential malware indicators.

Or you can visit the Virus total website for manual and deep analyses.



In real-world scenarios, you might encounter suspicious file names like "*Bonus Policy.pdf*" or "*Security Rules.exe.pdf*" that warrant further investigation. Yara rules can be a valuable tool for such situations. In our example, knowing the keywords "*message*" and "*PWNED*" can help us identify potentially malicious files. We can use the **Windows.Search.Yara** artifact with these keywords to search for EXE files containing the Unicode string "PWNED," as demonstrated below.



Windows.Search.Yara

Rule	HitOffset	HitContext	FileName	Size
Hit	5644	PWNED	\\.\C:\$Recycle.Bin\S-1-5-21-3802347957-1837500957-2111429085-1000\$RBXRQ52\pup4\pup4message.exe	10240
Hit	5644	PWNED	\\.\C:\pup4message.exe	10240

In conclusion, investigating PUPs with Velociraptor is a crucial step in staying safe and secure online. With the right tools and knowledge, we can identify and eliminate potential threats before they cause any harm.

Note:
Ethical considerations: *While this demonstration utilizes a* ***sample PUP*** *for educational purposes,* ***intentionally infecting a system with malware****, even for educational purposes, is* ***not recommended****. I encourage exploring alternative methods like using publicly available samples from reputable sources to showcase Velociraptor's capabilities without compromising real systems.*
Disclaimer: *This blog post is intended for* ***educational purposes only*** *and should not be used to harm or compromise real systems.*