El Jefe 1.1: The Boss Will See You Now

By Russ McRee – ISSA member, Puget Sound (Seattle), USA Chapter

Join the Discussion
Connect

Prerequisites

Virtualization platform or Linux OS with Python/Django

I've always been a fan of Windows process-oriented tools such as Process Explorer, Process Hacker, WinPatrol, and Process Monitor. Imagine combining process monitoring functionality with a client/server architecture that provides detailed information about process creation, privileges, and children, all stored and categorized via central logging mechanism.

Make way for El Jefe (the boss), a solution from Immunity that intercepts native Windows API process creation calls, allowing you to track, monitor, and correlate process creation events. Going many steps beyond tracking simple process creation, El Jefe provides a microscopic view of the binaries that are run: SHA1, PID, flags (oh my!), sorted chronologically with spawned offspring while clickable for instant analysis. Schwing! I haven't had this much fun infecting test VMs in months. If El Jefe was a doctor, he'd likely practice gastroenterology ("this may be a bit uncomfortable").

El Jefe writes to a centralized logging server, storing data in a PostgreSQL database by default, data which can be mined via a Django-born web application. El Jefe is inquisitive enough to provide a reasonable modicum of intrusion pattern correlation on your monitored hosts.

El Jefe's developer/project lead is Justin Seitz, and it's provided by Dave Aitel's Immunity, the same folks who offer CANVAS and the Immunity Debugger (both are excellent).

Justin hooked up a brother in a pinch as your author was scrambling to meet an accelerated deadline, ensuring the *Journal's* whip-cracking editor (Hi Thom ☺) happiness in advance of RSA 2011.

From Justin:

"With El Jefe, we basically wanted to take a look at the life cycle of an attack from a different perspective, namely process creation. We simply wanted to see what an attack looked like based solely on what processes were spawned and their varying levels of privilege. By beginning to explore this data we were able to pick out certain patterns (for example a SYSTEM process being started from Internet Explorer) that clearly indicated successful compromise. El Jefe not only preserves data for forensics use, including file paths and hashes, but also retrieves a full page of code from the entry point of the executed binaries. This raw code can be useful for reverse engineers or malware analysts to see what the first steps of execution look like for that binary."

When I asked Justin what his favorite use case might be, he said, "get El Jefe installed on a host that has some known vulnerabilities. Nail it with your favorite exploit framework, do some local privilege escalation attacks, drop some Trojans and any other nefarious behavior you can think of. Log in to the El Jefe web app and begin tracing through the events to see what your attack looks like. Hit our 'Intrusion' tab in the web app and you may be surprised at what we do and don't catch!"

Roger that, doing so now…

Paying dues to El Jefe

Server setup

Download the Ubuntu-based El Jefe VM appliance or roll your own from source.[1]

If you build from source consider doing so on an Ubuntu instance as the documentation favors it.

I'll bucket up a quick install method if you choose that path but keep in mind the VM works instantly and is the easier path.

Assuming you don't already have these dependencies installed, execute `sudo apt-get install postgresql-8.4 pgadmin3 python-psycopg2 python-openssl`. Grab the latest Django,[2] the Python Web framework, (1.2.4 is current as I write this), and install it:

```
tar zxvf Django-1.2.4.tar.gz
cd Django-1.2.4
sudo python setup.py install
```

Create a web app login for the El Jefe client to use via http://localhost/admin on your El Jefe server; this is the Django admin portal. Again, keep it simple and use *eljefe* for username and password to get started.

Via `psql` (command prompt) or `pgadmin3` (GUI), create a new login role that should be the same name as the user who will be running the web application (the VM uses *eljefe* exclusively).

1 http://eljefe.immunityinc.com.

2 http://www.djangoproject.com/download.

Create a new database, also named *eljefe*, and set the owner to be the login user you just created.

The source, when unpacked, includes a directory named *webapp*. Copy or move it to your `/var` directory. Edit the database settings as you configured them above via `/var/webapp/settings.py` then sync the database by running `python manage.py syncdb`.

Browse binaries

Page 1 of 1.

Search Results : 13

Station	Path	Binary SHA1
HIOMALWAREVM01	C:/Program Files/Wireshark/wireshark.exe	ab1d4a731dc8c89c584ace4141cc5f127c94f3a6
HIOMALWAREVM01	C:/WINDOWS/explorer.exe	9d2bf84874abc5b6e9a2744b7865c193c08d362f
HIOMALWAREVM01	C:/Program Files/Wireshark/dumpcap.exe	ba86a78f23b5c72bc9415d8242fbbc6bf6dc5e46
HIOMALWAREVM01	C:/WINDOWS/system32/cmd.exe	811a005cf787c6ccbe0d9f1c36c1d49a9cb71fd1

Figure 1 – El Jefe's binary browse view

Page 1 of 1.

Search Results : 9

Date	Parent Binary	Binary
2011-01-20 22:31:14.725000	C:/WINDOWS/explorer.exe	C:/Program Files/Microsoft Office/Office10/WINWORD.EXE
2011-01-20 22:31:15.115000	C:/WINDOWS/system32/services.exe	C:/WINDOWS/system32/svchost.exe
2011-01-20 22:31:30.334000	C:/WINDOWS/system32/svchost.exe	C:/WINDOWS/system32/dumprep.exe
2011-01-20 22:31:58.584000	C:/WINDOWS/system32/dumprep.exe	C:/WINDOWS/system32/dwwin.exe
2011-01-20 22:34:03.412000	C:/WINDOWS/explorer.exe	C:/Program Files/Internet Explorer/iexplore.exe
2011-01-20 22:34:05.662000	C:/Program Files/Internet Explorer/iexplore.exe	C:/Program Files/Internet Explorer/iexplore.exe
2011-01-20 22:34:26.162000	C:/Program Files/Internet Explorer/iexplore.exe	C:/WINDOWS/system32/rundll32.exe
2011-01-20 22:34:41.897000	C:/WINDOWS/explorer.exe	C:/malware/bifrost.rootkit/photohi5.exe
2011-01-20 22:47:50.506000	C:/WINDOWS/explorer.exe	C:/iDEFENSE/MAP/ShellExt.exe

Page 1 of 1.

Figure 2 – El Jefe's event browser view

Create a web app login for the client to use via http://localhost/admin on your El Jefe server; this is the Django admin portal. Again, keep it simple and use *eljefe* for username and password to get started.

Finish up by first starting xmlserver via `cd /var/webapp/xmlserver && python ElJefeXMLServer.py` then the web application via `cd /var/webapp/ && python manage.py runserver localhost:8000`.

Either way you opt to implement your El Jefe server, utilize the user's manual as reference.[3]

Client setup

On your Windows hosts, download the client bits, unpack them, and initiate the appropriate `.msi` file (32bit or 64bit).

The server configuration window will present itself; give it your El Jefe server IP address, server port (default for the XML server is 5555) and the *eljefe* username and password created above.

Note: I encountered a bug and worked through it before reading the forum[4] where it was so readily clarified. Avoid my pain and idiocy by being aware that sometimes no data will be sent from the client host to the XML server if *MSVCR71.dll* is not present in `C:\Windows\System32`. You can confirm client failure if you're not seeing data via the web UI by changing directories to the El Jefe client install path, `C:\Program Files\Immunity\El Jefe` for Windows on Windows XP 32-bit, and issuing `eljefeSrv.exe –application –debug`. If you see an error alert box for *MSVCR71.dll*, you know what to do. I grabbed a copy from the *IR\Imager* directory on the Helix distribution I carry in my incident response jump bag.

Whew, installation and setup complete. Let's put El Jefe through his paces.

3 http://eljefe.immunityinc.com/Release-1.1/El%20Jefe%20Users%20Guide.pdf.

4 https://forum.immunityinc.com/board/thread/1359/bugs-found-in-el-jefe-beta-3-0/?page=1#post-1359.

El Jefe has your back

El Jefe's UI will offer you tabbed options including *Stations* (your monitored hosts), *Binaries* (monitored processes), *Events* (events sorted chronologically), *Intrusion* (queried by date and method; more on this later), *Data* (DROP ALL will do just that, you've been warned), and *Docs* which includes the user's guide. All views include search or browse functionality. Figure 1 shows my malware testing VM in a known good state prior to infection, via the Browse binaries view. You'll appreciate the fact this view includes the SHA1 hash for search engine research.

Time for some fun!

Perhaps you've heard of or encountered a Trojan known as Win32.Poison, CeeInject, Buzus, or Bifrost.

The last time I reviewed Bifrost, I noted an interesting result during strings analysis, a result that El Jefe renders admirably.

Figure 2 shows binary events chronologically in a sequence, albeit simplified, that you might expect from a typical user. User fires up the system, opens Word, it crashes, preps a dump, user fires up IE, user stomps malware.

Binary Information

Hostname	HIOMALWAREVM01
IP Address	192.168.248.15
Full Path	C:/malware/bifrost.rootkit/photohi5.exe
Binary SHA1	966cbd542ffb2e5753c5b9f24dccb55aeffaa992
Binary Size	59954
Architecture	32 bit executable
Code Section SHA1	40920a29cb71a6011e8a08c6724f8184fcace49b
Code Entropy	1.78288323567
Code Section	00dcebfdfbe3ebda .... e6fcebefea000000 .... 00e5ebfce0ebe2bd .... bca0eae2e2000000 .... 00e0faeae2e2a0ea .... e2e20000006e6173 .....nas 7479000000010000 ty...... 0064626768656c70 .dbghelp 2e646c6c00536269 .dll.Sbi 65446c6c2e646c6c eDll.dll

Figure 3 – El Jefe's event browser view

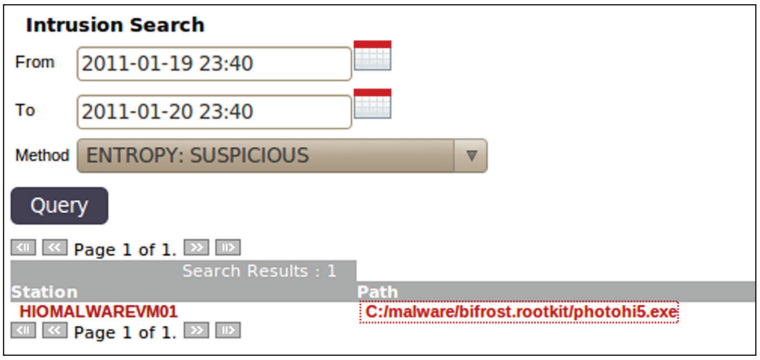

Figure 4 – El Jefe confirms intrusion by method

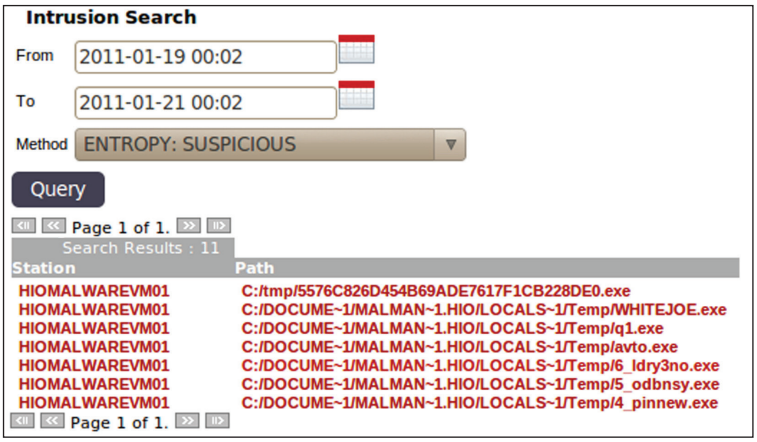

Figure 5 – El Jefe "suspects" Zeus

Note `photohi5.exe` in Figure 3. As El Jefe features click through drill down, I clicked the photohi5.exe-related event. Arriving at Binary Information I reviewed the Code Section (HEX and ASCII) and found my favorite string reference: "Nasty."

Yep.

Note the Code Entropy reference in Figure 4. Let's search *Intrusion* methods by *Entropy: Suspicious*. Bingo, Figure 4 confirms.

Nothing fancy, but effective yes? Not convinced? Figure 5 offers suspicious entropic detection of Zeus unloading itself on my victim system.

I wanted to track behavior from my Windows Server 2003 Pwnzor Edition VM (it's entirely unpatched and therefore offers endless paths for experimentation) as well. I loaded the El Jefe client then hit the VM with Metasploit. I chose an SMB attack exploiting "a parsing flaw in the path canonicalization code of NetAPI32.dll through the Server Service" (MS08-067). I followed exploitation by firing up a Meterpreter shell, a command shell, then goofing with `lsass.exe` in a few different ways. I was hopeful that I'd trigger an Intrusion alert based on `lsass.exe` as the executing parent and wasn't successful, but the El Jefe event browser exhibited my actions chronologically with reasonable success as seen in Figure 6.

The first reader who emails me and can tell me specifically what I did, based on what you see in Figure 6 will receive an information security book.

In conclusion

When I have this much fun preparing the month's column, even under pressure, you know the tool is worthy of your time. As Justin mentioned in his feedback, during testing El Jefe will "provide feedback" on your malfeasance, even if you wish to avoid playing with live malware. Give a vulnerable test system a good poke with Metasploit (or CANVAS if you're a customer ☺) and prepare to be tracked.

You can assume that properly configured, El Jefe can serve you in a production capacity, perhaps not on an enterprise scale, but likely effective in SMB shops.

Send feedback and bugs to the El Jefe team via eljefe at imunityinc dot com or the forum.[5]

Cheers…until next month.

Acknowledgements

—Justin Seitz, El Jefe developer and project lead.

About the Author

Russ McRee, GCIH, GCFA, GPEN, CISSP, is team leader and senior security analyst for Microsoft's Online Services Security Incident Management team. As an advocate of a holistic approach to information security, Russ' website is holisticinfosec.org. Contact him at russ@holisticinfosec.org.

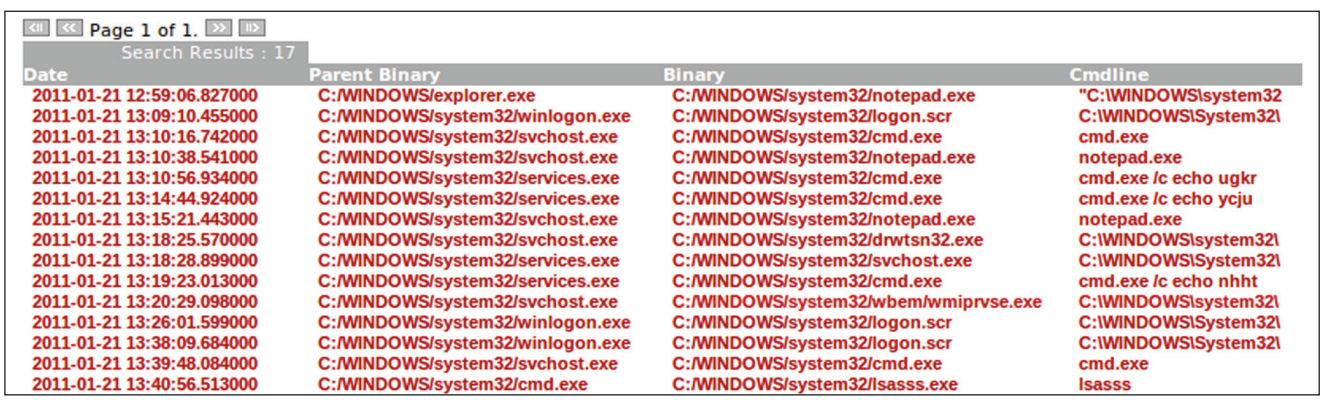

Figure 6 – El Jefe Metasploit generated events

5 https://forum.immunityinc.com/board/show/0/.