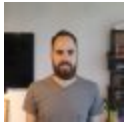


# New Variants of Agent.BTZ/ComRAT Found: The Threat That Hit The Pentagon In 2008 Still Evolving; Part 1/2

 [intezer.com/new-variants-of-agent-btz-comrat-found/](https://intezer.com/new-variants-of-agent-btz-comrat-found/)

August 7, 2017



Written by Omri Ben Bassat - 7 August 2017



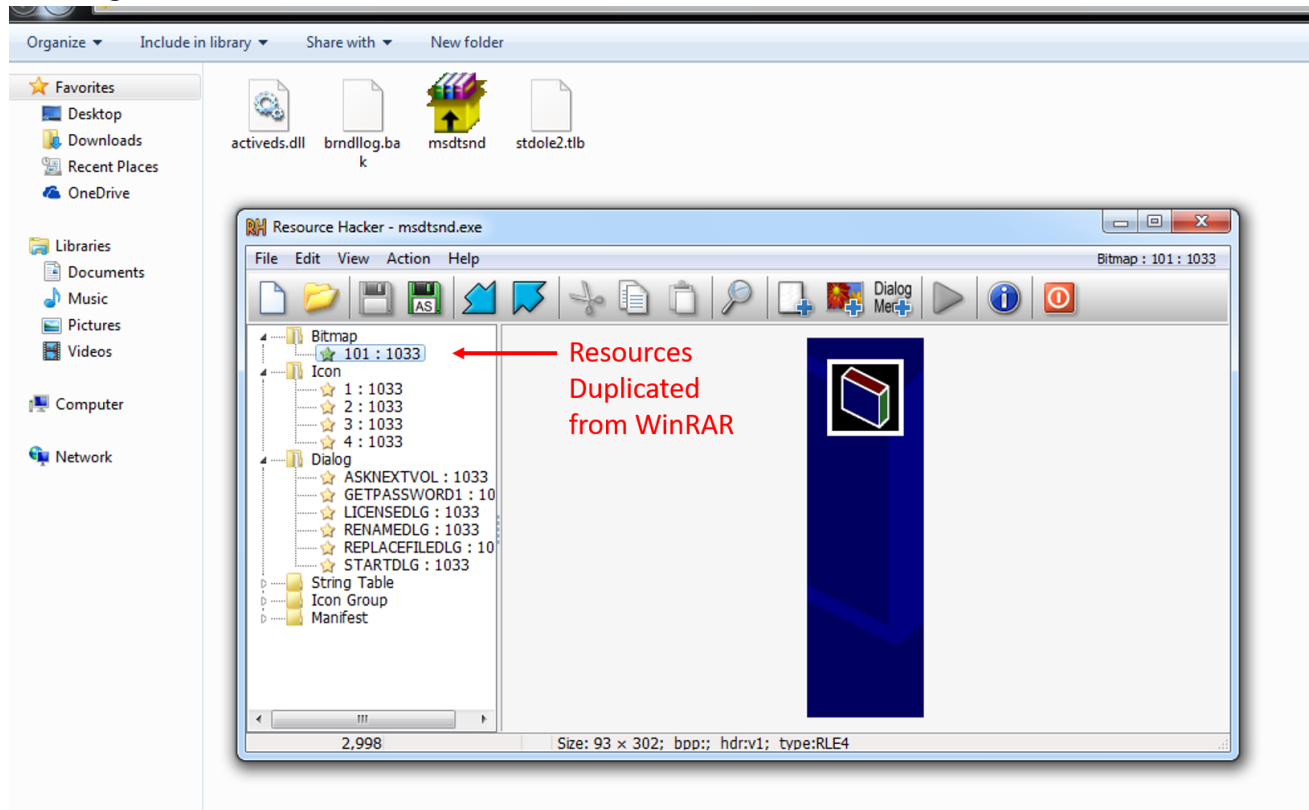
## [Get Free Account](#)

[Join Now](#)

Agent.BTZ—also known as ComRAT—is one of the world’s oldest known state-sponsored threats, mainly known for the 2008 Pentagon breach. Technically speaking, Agent.BTZ is a sophisticated user-mode RAT developed and operated by the Turla group in conjunction with Snake/Uroburos rootkit. In the past few months, we conducted research on Agent.BTZ’s code-base and how it evolved using Intezer Code Intelligence™ technology. **Based on our research conclusions, we were able to hunt about a dozen new samples and more than seventy previously unknown live IP & DNS addresses indicating the ongoing abuse of satellite internet providers operating in both Africa & the Middle East.**

This is a short memo regarding our findings from the past few months; in a few days, we will publish a whitepaper (part 2/2) describing in more details how we found these new variants using our technology, along with a thorough analysis of the new samples.

**Dropper:** Although the code itself was written from scratch and has nothing to do with WinRAR, the adversary tried to mimic WinRAR's SFX installer. Resource data was duplicated, including icons and layouts used by the original installer, as you can see in the following screenshot:



Once executed, the dropper installs activeds.dll – a proxy dll which is loaded directly to explorer.exe once the machine reboots. The purpose of this proxy dll is to load the malware’s main payload – stdole2.tlb. The dropper then also deletes any previous installation of Agent.BTZ if it exists. This is done using a hard-coded file path:

- **“C:\Documents and Settings\<USER>\Application Data\Microsoft\Windows\Themes\termsvr32.dll”**
- **“C:\Documents and Settings\<USER>\Application Data\Microsoft\Windows\Themes\pcasrc.tlb”**

*\*\*Note: These file names were first used by Agent.BTZ in late 2014, as you can see in this [automatic Dr. WEB report](#)*

Once finished, the dropper renames and self delete using the following command line:

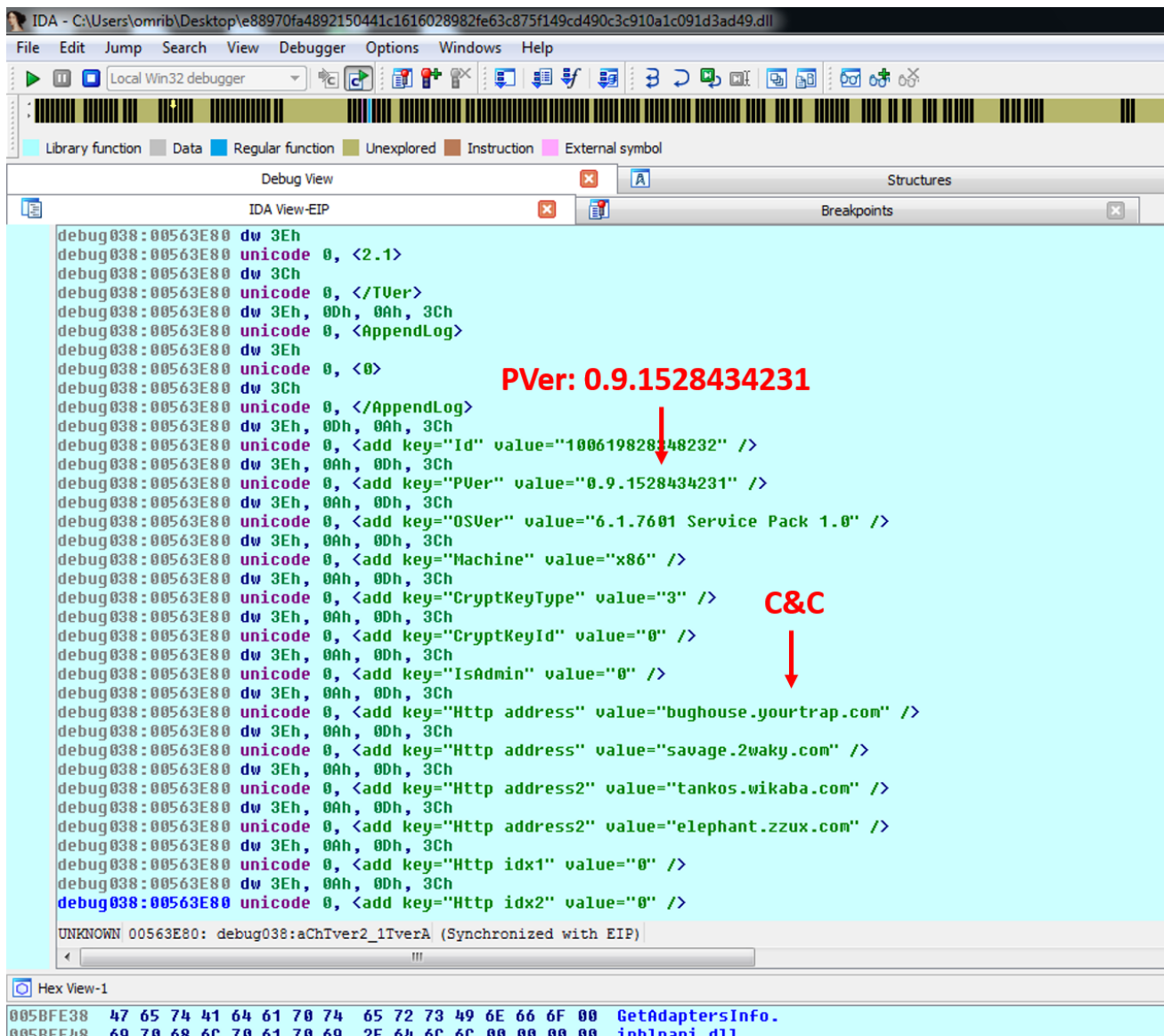
```
“C:\WINDOWS\system32\rundll32.exe C:\DOCUME~1<USER>~1\APPLIC~1\MICROS~1\Windows\stdole2.tlb,UnInstall C:\~$.tmp”
```

Samples found:

1. 69690f609140db503463daf6a3699f1bf3e2a5a6049cefe7e6437f762040e548

## 2. 6798b3278ae926b0145ee342ee9840d0b2e6ba11ff995c2bc84d3c6eb3e55ff4

**stdole2.tlb:** As previously mentioned, this file is the main component installed by the fake-sfx dropper and loaded by activeds.dll. We extracted the configuration from each sample in order to obtain the c2 address and inner version (“PVer”), which is built into every Agent.BTZ sample. In the past, Agent.BTZ’s developers have used an incremental value to indicate the inner build version, the last known value is 3.26 as published by G-Data in late 2014. It seems that the developers have reacted to G-Data’s publication and have stopped using an incremental value. New variants are now using a different numbering system of 0.8/9.<RANDOM\_VALUE>, making it more difficult for researchers to track the exact version of the samples.



The screenshot shows the IDA Pro interface with the following assembly code and annotations:

```
debug038:00563E80 dw 3Eh
debug038:00563E80 unicode 0, <2.1>
debug038:00563E80 dw 3Ch
debug038:00563E80 unicode 0, </TVer>
debug038:00563E80 dw 3Eh, 0Dh, 0Ah, 3Ch
debug038:00563E80 unicode 0, <AppendLog>
debug038:00563E80 dw 3Eh
debug038:00563E80 unicode 0, <0>
debug038:00563E80 dw 3Ch
debug038:00563E80 unicode 0, </AppendLog>
debug038:00563E80 dw 3Eh, 0Dh, 0Ah, 3Ch
debug038:00563E80 unicode 0, <add key="Id" value="100619828148232" />
debug038:00563E80 dw 3Eh, 0Ah, 0Dh, 3Ch
debug038:00563E80 unicode 0, <add key="PVer" value="0.9.1528434231" />
debug038:00563E80 dw 3Eh, 0Ah, 0Dh, 3Ch
debug038:00563E80 unicode 0, <add key="OSVer" value="6.1.7601 Service Pack 1.0" />
debug038:00563E80 dw 3Eh, 0Ah, 0Dh, 3Ch
debug038:00563E80 unicode 0, <add key="Machine" value="x86" />
debug038:00563E80 dw 3Eh, 0Ah, 0Dh, 3Ch
debug038:00563E80 unicode 0, <add key="CryptKeyType" value="3" />
debug038:00563E80 dw 3Eh, 0Ah, 0Dh, 3Ch
debug038:00563E80 unicode 0, <add key="CryptKeyId" value="0" />
debug038:00563E80 dw 3Eh, 0Ah, 0Dh, 3Ch
debug038:00563E80 unicode 0, <add key="IsAdmin" value="0" />
debug038:00563E80 dw 3Eh, 0Ah, 0Dh, 3Ch
debug038:00563E80 unicode 0, <add key="Http address" value="bughouse.yourtrap.com" />
debug038:00563E80 dw 3Eh, 0Ah, 0Dh, 3Ch
debug038:00563E80 unicode 0, <add key="Http address" value="savage.2waky.com" />
debug038:00563E80 dw 3Eh, 0Ah, 0Dh, 3Ch
debug038:00563E80 unicode 0, <add key="Http address2" value="tankos.wikaba.com" />
debug038:00563E80 dw 3Eh, 0Ah, 0Dh, 3Ch
debug038:00563E80 unicode 0, <add key="Http address2" value="elephant.zzux.com" />
debug038:00563E80 dw 3Eh, 0Ah, 0Dh, 3Ch
debug038:00563E80 unicode 0, <add key="Http idx1" value="0" />
debug038:00563E80 dw 3Eh, 0Ah, 0Dh, 3Ch
debug038:00563E80 unicode 0, <add key="Http idx2" value="0" />
```

Annotations in the image:

- PVer: 0.9.1528434231** (red text) with a red arrow pointing to the line: `unicode 0, <add key="PVer" value="0.9.1528434231" />`
- C&C** (red text) with a red arrow pointing to the line: `unicode 0, <add key="Http address" value="bughouse.yourtrap.com" />`

Hex View-1 at the bottom shows:

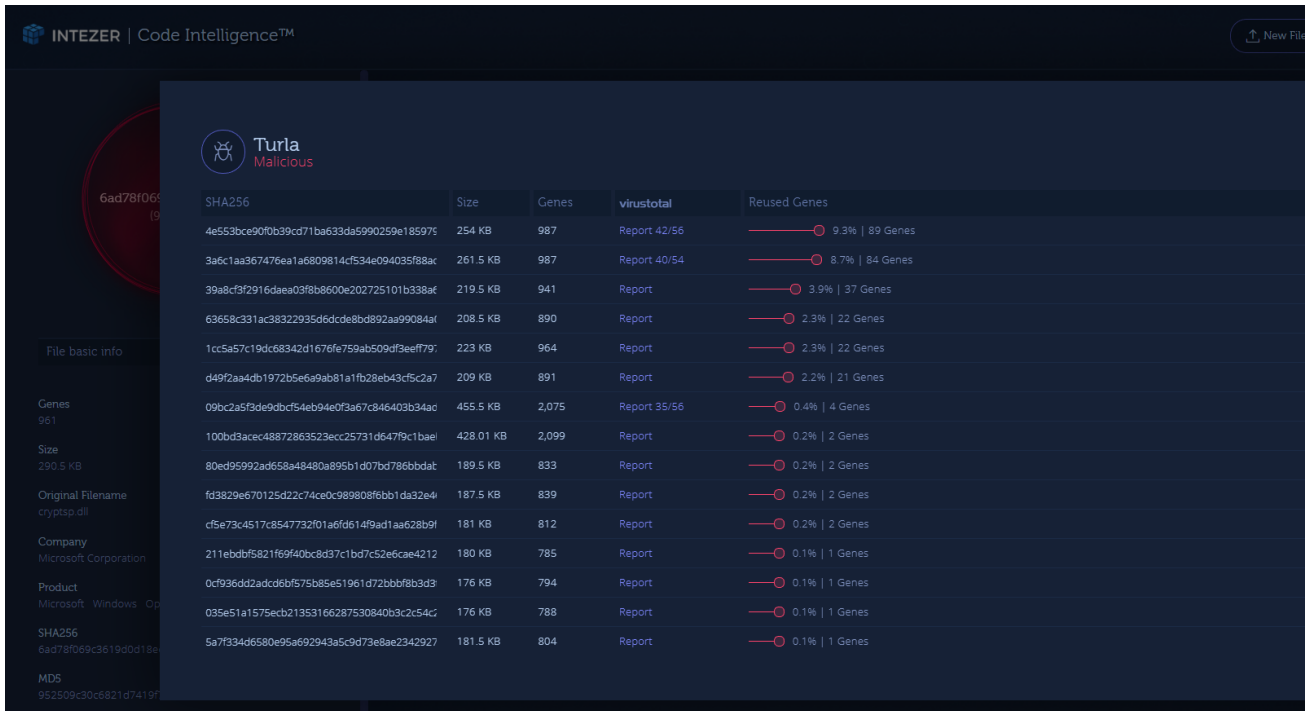
```
005BFE38 47 65 74 41 64 61 70 74 65 72 73 49 6E 66 6F 00 GetAdaptersInfo.
005BFE48 60 70 68 6C 70 61 70 60 2F 64 6C 6C 00 00 00 00 inhlpapi.dll
```

\*\*example configuration extracted from one of the samples – PVer 0.9.1528434231.

Even without the PVer numbering, we were able to determine using our technology that these samples are from a newer version, which is based on the latest known versions of Agent.BTZ – 3.25 / 3.26. These are the two top files you can see in the following screenshot:

1. [4e553bce90f0b39cd71ba633da5990259e185979c2859ec2e04dd8efcdafe356\(VirusTotal\)](#)
2. [3a6c1aa367476ea1a6809814cf534e094035f88ac5fb759398b783f3929a0db2\(VirusTotal\)](#)

Both of these files were uploaded almost three years ago to VT(!)



\*\*A screenshot from the Intezer Analyze™ product displaying a list of files in our database that share pieces of code with one of the new samples. These pieces of code are specific to the Turla malware family, and were not seen in any other malicious or legitimate software.

Samples found:

1. 6ad78f069c3619d0d18eef8281219679f538cfe0c1b6d40b244beb359762cf96
2. 49c5c798689d4a54e5b7099b647b0596fb96b996a437bb8241b5dd76e974c24e
3. e88970fa4892150441c1616028982fe63c875f149cd490c3c910a1c091d3ad49
4. 89db8a69ff030600f26d5c875785d20f15d45331d007733be9a2422261d16cea

**Indicators of Compromise:**

type	indicator
sha256	69690f609140db503463daf6a3699f1bf3e2a5a6049cefe7e6437f762040e548
sha256	6798b3278ae926b0145ee342ee9840d0b2e6ba11ff995c2bc84d3c6eb3e55ff4
sha256	73db4295c5b29958c5d93c20be9482c1efffc89fc4e5c8ba59ac9425a4657a88
sha256	50067ebcc2d2069b3613a20b81f9d61f2cd5be9c85533c4ea34edbefaeb8a15f
sha256	380b0353ba8cd33da8c5e5b95e3e032e83193019e73c71875b58ec1ed389bdac
sha256	9c163c3f2bd5c5181147c6f4cf2571160197de98f496d16b38c7dc46b5dc1426
sha256	628d316a983383ed716e3f827720915683a8876b54677878a7d2db376d117a24
sha256	f27e9bba6a2635731845b4334b807c0e4f57d3b790cecdc77d8fef50629f51a2
sha256	a093fa22d7bc4ee99049a29b66a13d4bf4d1899ed4c7a8423fbb8c54f4230f3c
sha256	6ad78f069c3619d0d18eef8281219679f538cfe0c1b6d40b244beb359762cf96
sha256	49c5c798689d4a54e5b7099b647b0596fb96b996a437bb8241b5dd76e974c24e
sha256	e88970fa4892150441c1616028982fe63c875f149cd490c3c910a1c091d3ad49
sha256	89db8a69ff030600f26d5c875785d20f15d45331d007733be9a2422261d16cea
ip	81.199.34[.]150
dns	elephant.zzux[.]com
dns	angrybear.ignorelist[.]com
dns	bigalert.mefound[.]com
dns	bughouse.yourtrap[.]com
dns	getfreetools.strangled[.]net
dns	news100top.diskstation[.]org
dns	pro100sport.mein-vigor[.]de
dns	redneck.yourtrap[.]com
dns	savage.2waky[.]com
dns	tehnologtrade.4irc[.]com
ip	81.199.160[.]11

---

dns	forums.chatnook[.]com
dns	goodengine.darktech[.]org
dns	locker.strangled[.]net
dns	simple-house.zzux[.]com
dns	specialcar.mooo[.]com
dns	sunseed.strangled[.]net
dns	whitelibrary.4irc[.]com
dns	bloodpearl.strangled[.]net
dns	getlucky.ignorelist[.]com
dns	proriot.zzux[.]com
dns	fourapi.mooo[.]com
dns	nopasaran.strangled[.]net
ip	78.138.25[.]29
dns	showme.twilightparadox[.]com
dns	mouses.strangled[.]net
ip	82.146.175[.]69
dns	mouses.strangled[.]net
ip	178.219.68[.]242
dns	ftp.fueldust.compress[.]to
dns	ftp.linear.wikaba[.]com
dns	ftp.mysterysoft.epac[.]to
dns	ftp.scroller.longmusic[.]com
dns	ftp.spartano.mefound[.]com
dns	fueldust.compress[.]to
dns	linear.wikaba[.]com
dns	mysterysoft.epac.to

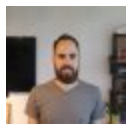
---

dns	safety.deaftone[.]com
dns	salary.flnet[.]org
dns	scroller.longmusic[.]com
dns	spartano.mefound[.]com
ip	88.83.25[.]122
dns	robot.wikaba[.]com
ip	41.223.91[.]217
dns	smileman.compress[.]to
dns	decent.ignorelist[.]com
dns	dekka.biz[.]tm
dns	disol.strangled[.]net
dns	eraser.2waky[.]com
dns	filelord.epac[.]to
dns	justsoft.epac[.]to
dns	smuggler.zzux[.]com
dns	sport-journal.twilightparadox[.]com
dns	sportinfo.yourtrap[.]com
dns	stager.ignorelist[.]com
dns	tankos.wikaba[.]com
dns	grandfathers.mooo[.]com
dns	homerich.mooo[.]com
dns	jamming.mooo[.]com
dns	pneumo.mooo[.]com
dns	razory.mooo[.]com
dns	anger.scieron[.]com
dns	gantama.mefound[.]com

dns	letgetbad.epac[.]to
dns	rowstate.epac[.]to
dns	memento.info[.]tm
ip	196.43.240[.]177
dns	bughouse.yourtrap[.]com
dns	news100top.diskstation[.]org
ip	169.255.102[.]240
dns	harm17.zzux[.]com
dns	mountain8.wikaba[.]com
sha256	0e0045d2c4bfff4345d460957a543e2e7f1638de745644f6bf58555c1d287286
sha256	bdcc7e900f10986cdb6dc7762de35b4f07f2ee153a341bef843b866e999d73a3
sha256	fac13f08afe2745fc441ada37120cebce0e0aa16d03a03e9cda3ec9384dd40f2
sha256	bae62f7f96c4cc300ec685f42eb451388cf50a13aa624b3f2a019d071fddaeb1

**Related articles:**

1. <https://www.gdatasoftware.com/blog/2014/11/23937-the-urobuos-case-new-sophisticated-rat-identified>
2. <https://www.gdatasoftware.com/blog/2015/01/23927-evolution-of-sophisticated-spyware-from-agent-btz-to-comrat>
3. <http://blog.threatexpert.com/2008/11/agentbtz-threat-that-hit-pentagon.html>
4. <https://securelist.com/satellite-turla-apt-command-and-control-in-the-sky/72081/>
5. [https://www.symantec.com/content/en/us/enterprise/media/security\\_response/whitepapers/waterbug-attack-group.pdf](https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/waterbug-attack-group.pdf)
6. <https://securelist.com/the-epic-turla-operation/65545/>
7. [http://artemonsecurity.com/snake\\_whitepaper.pdf](http://artemonsecurity.com/snake_whitepaper.pdf)
8. <https://www.gdatasoftware.com/blog/2015/01/23926-analysis-of-project-cobra>



**Omri Ben Bassat**

Ex-officer in the IDF CERT. Malware analyst and a reverse engineer with vast experience in dealing with nation-state sponsored cyber attacks. Omri is the creator of Master of Puppets (MoP)—an open-source framework for reverse engineers who wish to create and operate



trackers for new malware found in the wild—which was presented during the Black Hat USA 2019 Arsenal.