ネットワーク設計/応用
演習 1 レポート

学籍番号 1270381 宮本武

2024 年 12 月 25 日

1　演習を通して学んだこと

1.1　演習内容

今回の演習では，サイバーセキュリティネクサス（CYNEX）のネットワーク内のサーバに工科大の個人端末からリモート接続を行うことで，OS コマンドインジェクションによる攻撃及び防御の手法を学んだ.

具体的な演習内容としては,

攻撃手法　CYNEX のネットワーク内にある講師用コンテンツサーバ（ドメイン検索サイト）に対して，同じネットワークに属すコンピュータからサイトにアクセスし，OS コマンドインジェクションを実行することで，サーバ内のユーザ名，パスワードといった機密情報を不正に取得する.

防御手法　OS コマンドインジェクションが実行される原因を排除し，健全なサーバ運営が行える様にプログラムを改良する,

であった.

1.2　インジェクション実行結果

講師用コンテンツサーバのドメイン検索サイトにて，ドメイン名を入力すべきフォームに

```
; cat <ファイルパス>
```

という形式で機密情報が入ったファイルのパスを指定してフォームを送信したところ，サーバからの返信欄にドメイン検索結果に続いて，機密情報が入ったファイルの内容が表示された.

1.3 危険性と原因

インジェクションの実行結果に機密情報が入ってしまっていることから，悪意を持ったユーザが上記のような手法を用いた場合，容易にサーバの機密情報を取得され，情報を悪用される可能性がある．このような情報漏洩を許してしまったのは，サイトのプログラムのソースコードに問題があったからである．

ドメイン検索はシェルの dig コマンドで行い，その標準出力を検索ページの結果欄に表示させるという仕組みであった．問題があったのは subprocess モジュールの run 関数によるシェル呼び出しの部分で，ユーザからの入力をドメイン名としてそのまま単一文字列のコマンドとしてシェルに渡していることで，想定外の出力結果を表示させることが可能になっている．シェルでは”;”（セミコロン）が改行と実質的に同じ意味を持ち，一行で複数のコマンドを書くことができる．セミコロンで区切った後半の”cat <ファイルパス>” も dig コマンドの後に実行され，機密情報が入ったファイルの内容が標準出力として，検索ページに表示されたのである．

1.4 対処方法

OS コマンドインジェクションを防ぐための方法は様々ある．今回の演習で扱った方法は subprocess モジュールの run 関数の引数を変更するという方法である．改良前には，shell=True という引数を指定していたが，これを shell=False にすることでシェルによる実行を防ぐことができる．cmd は単一文字列ではなく，文字列のリストとして指定する必要があるが，セミコロンを使用して別のプログラムを実行されることを防ぐことができる．

2 感想

実際に自分の手でサーバへ攻撃してみるという経験は無かったため，今回の演習は非常に興味深いものであった．しかし，攻撃を行う際の環境が複雑で，演習中は自分が何をしているのか，分からないことがあったのが残念である．

参考文献

[1] Python 3.13 標準ライブラリ サブプロセス管理 https://docs.python.org/ja/3.13/library/subprocess.html

[2] Bash Reference Manual https://www.gnu.org/software/bash/manual/bash.html