RSAConference2019
San Francisco | March 4–8 | Moscone Center


SESSION ID: HT-W03

# How Vault 7 Leaks Helped Develop My Own Cyber Espionage Weapon

**Wayne Ronaldson**

Red Team Lead
Loop Secure
@loop_secure

The difference between script kiddies and professionals is the difference between merely using other people's tools and writing your own.

"Charlie Miller

# Vault Seven Leaks

- AfterMidnight.
- Assassin.
- Athena.
- Hera.
- Pandemic.
- Grasshopper.
- Marble Framework.
- Wolfcreek.

# Vault 7: Projects


This publication series is about specific projects related to the Vault 7 main publication.

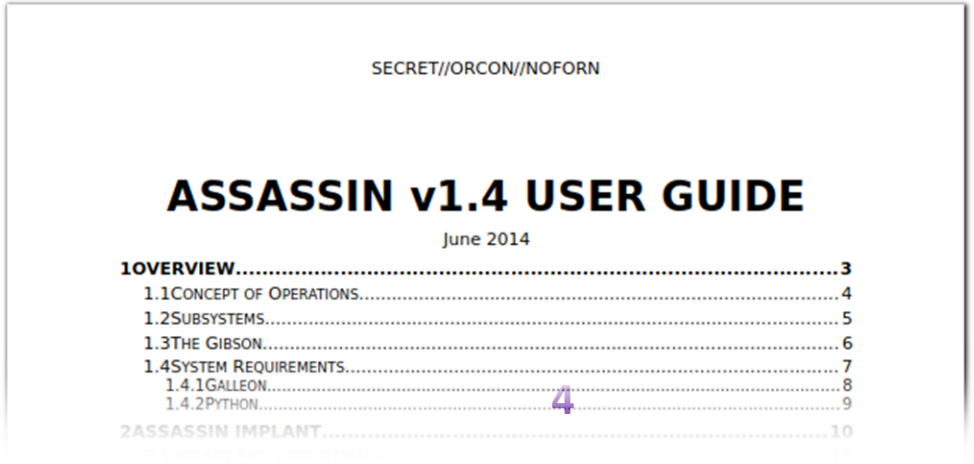
Releases ▾ Documents ▾

AfterMidnight

## Assassin v1.4 Users Guide

1 June, 2014

1 2 3 … 203 204


SECRET//ORCON//NOFORN

ASSASSIN v1.4 USER GUIDE

June 2014



4

### Downloads

Assassin_v1_4_Users_Guide.pdf

# Concept of Operations

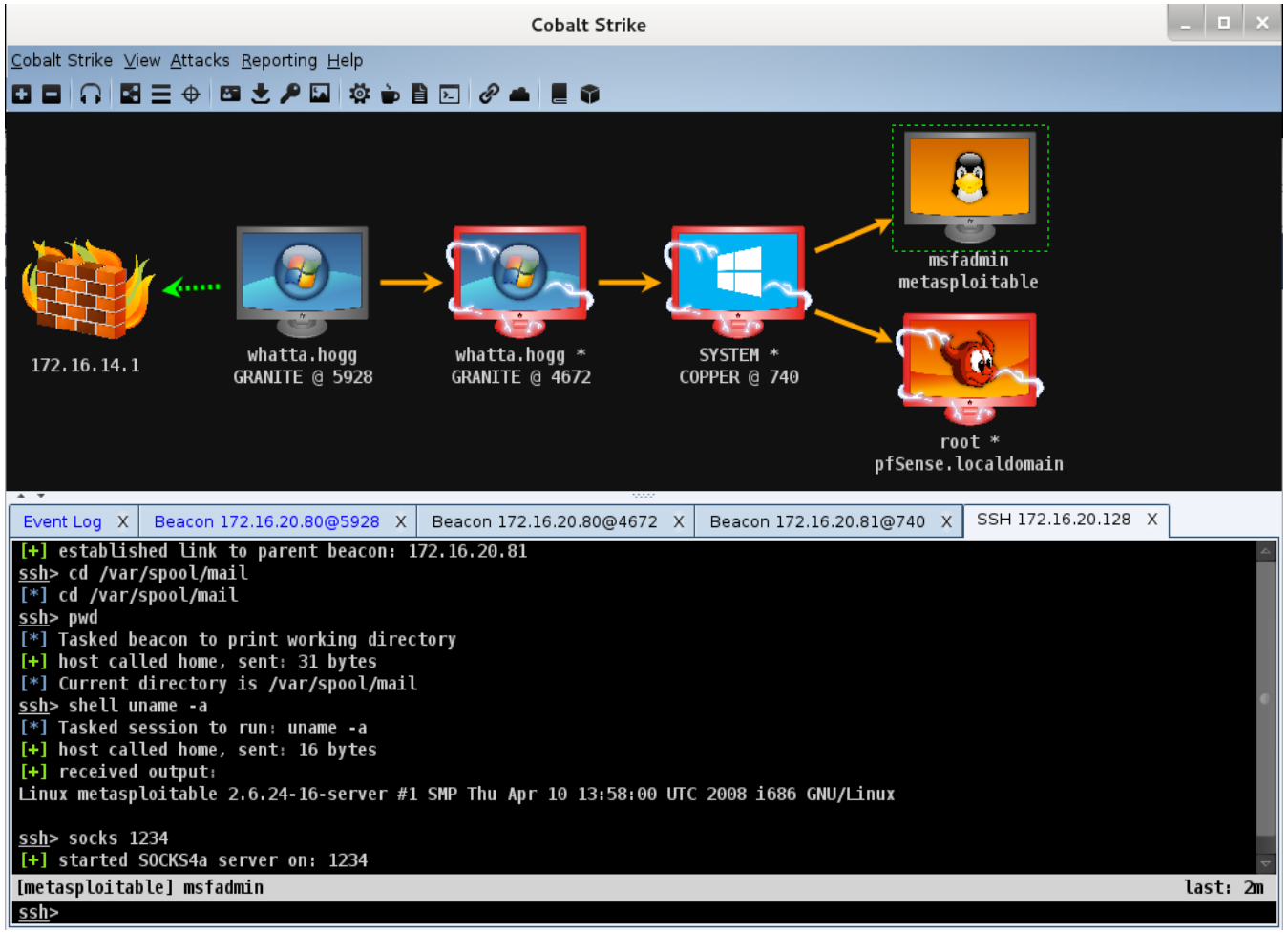
SECRET//ORCON//NOFORN

## 1.1 Concept of Operations

Assassin is an automated Implant that provides a simple collection platform on remote computers running the Microsoft Windows operating system. Once the tool is installed on the target, the implant is run within a Windows service process. Assassin will then periodically beacon to its configured listening post(s) to request tasking and deliver results. Communication occurs over one or more transport protocols as configured before or during deployment.

# Why Command & Control

- Super awesome.
- Cobalt Strike, Canvas, Empire.
- Metasploit, Core Impact.
- State sponsored adversary.
- Digital espionage.
- Want long term implant.
- Only for high value targets.
- Mr. Robot stole one.
- This is my digital Everest.

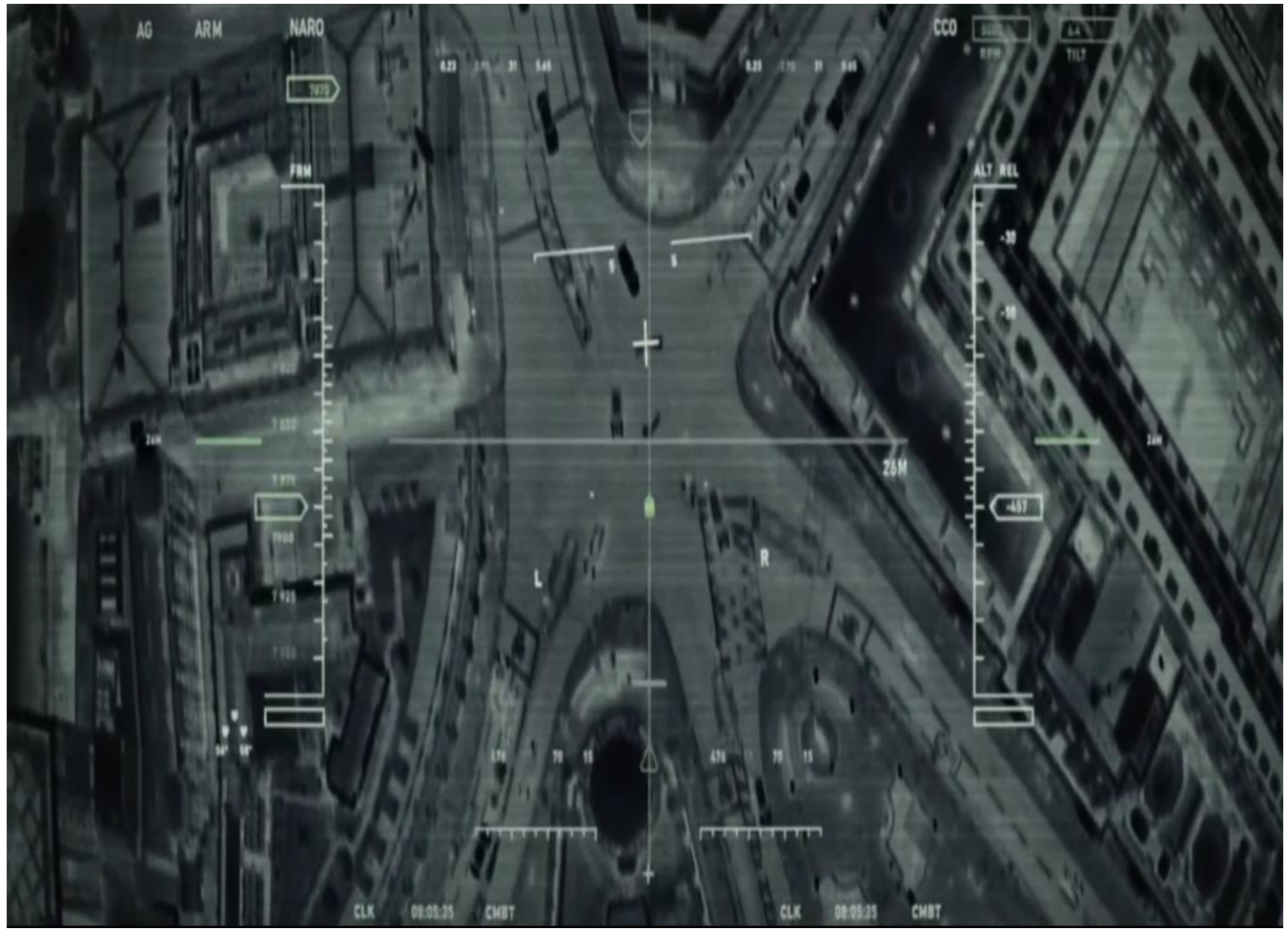


# Command & Control Wish List

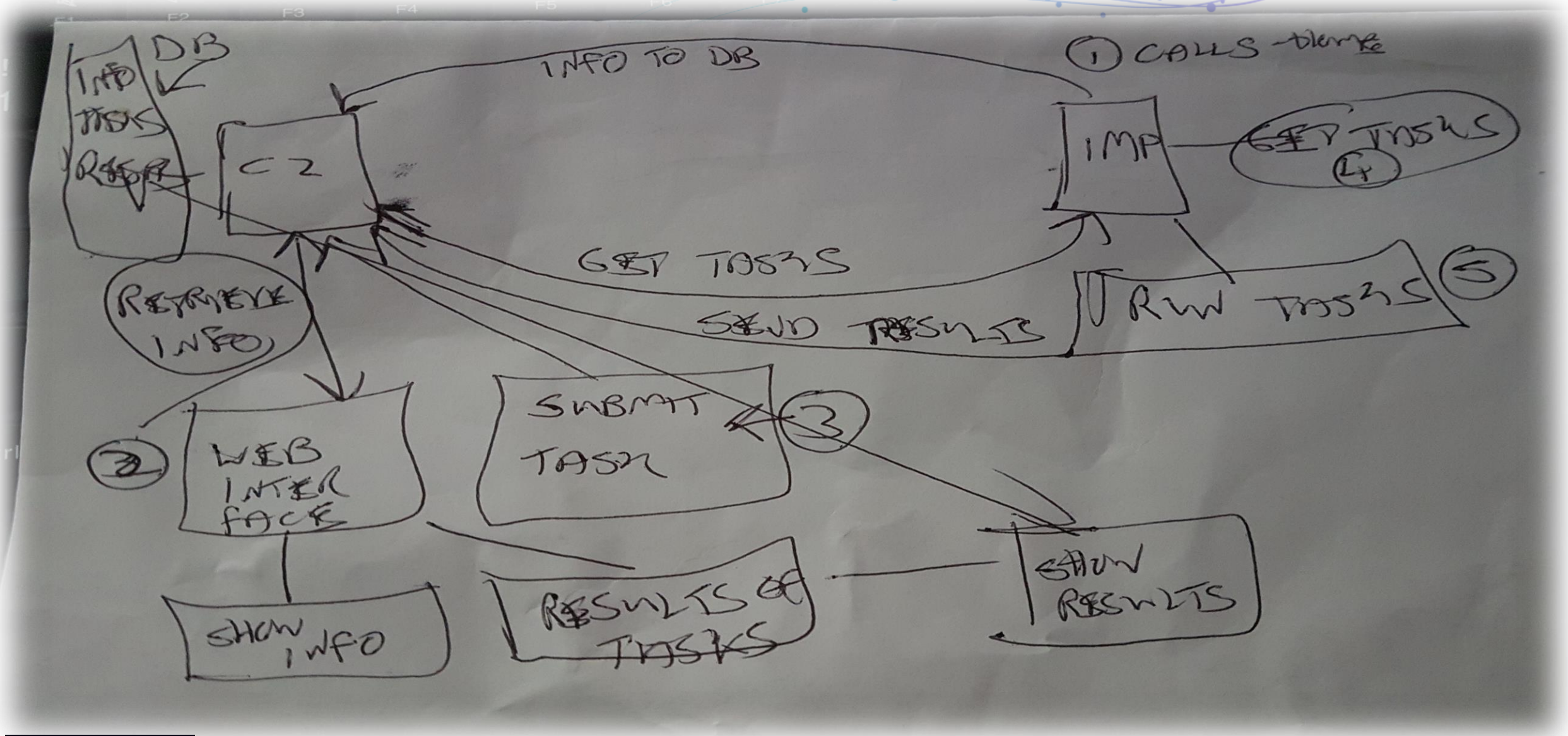
- Access easily.
- Quick setup.
- Secure communication.
- Dashboard view.
- Easy to operate.
- Task driven.
- Database.
- Scalable.

# Implant Wish List

- DLL, EXE.
- Task execution.
- User land.
- Audio capture.
- Screen capture.
- Keylogger.
- Memory based.
- Secure communication.



# Sunday 15th October 2017
# Day One


DB
INFO TO DB
1 CALLS
C2
IMP
GET TASKS
4
GET TASKS
SEND RESULTS
RUN TASKS
5
RETRIEVE INFO
2
WEB INTERFACE
SUBMIT TASK
3
SHOW INFO
RESULTS OF TASKS
SHOW RESULTS

# Implant Development

- RESTful API.
- Python.
- C++.
- Post request.
- Get request.
- Beacon time.
- Encryption.

# Welcome!

The C++ REST SDK is a Microsoft project for cloud-based client-server communication in native code using a modern asynchronous C++ API design. This project aims to help C++ developers connect to and interact with services.

# Getting Started

Vcpkg package 2.10.6

Homebrew package 2.10.7

Ubuntu 18.04 package 2.10.2

Fedora Rawhide package 2.10.7

openSUSE Tumbleweed package 2.10.6

Debian Testing package 2.10.7

Azure Pipelines failed

# Command & Control Development

- Python Flask.
- RESTful API.
- Ubuntu server.
- PostgreSQL.
- Web user interface.
- Task commands.
- Easy to deploy.
- Lightweight.

# Tasking

- Asynchronous.
- User interface.
- Task commands.
- Callback period.
- Stored database.
- Process tasks.
- Task results.

# Communication

- Fetch tasks.
- HTTPS.
- Java Web Token.
- Proxy auto detect.
- Upload.
- Download.

# Operational Window

- When implant is active.
- Hibernate time.
- Before going active.
- Hibernation period.
- New JWT.
- JWT expiry.
- Callback period.

# User Interface

- Manage implants.
- Generate tasks.
- Targeting HVTs.
- Implant registration.
- List tasks.
- View results.
- Task commands.

# System Survey

- OS info.
- Network info.
- Mounted drives.
- Current process.
- Drivers.
- Installed software.
- Services.
- PSP.
- Persistence.
- Audit.
- Scheduled tasks.
- Recently modified files.
- Recent USB devices.
- Passwords.
- Browsers.
- DNS cache.
- Recent RDP sessions.
- Shares.

# Vault 7: CIA Hacking Tools Revealed


Releases ▾ Documents ▾

Navigation: » Directory » AED Development Tradecraft » AED Development Tradecraft Home

Owner: User #2064619

## Development Tradecraft DOs and DON'Ts

SECRET//NOFORN

## (U) General (e.g. all PE/Mach-O/ELF or other binary files)

| Directive | Rationale |
|---|---|
| (S//NF) DO obfuscate or encrypt all strings and configuration data that directly relate to tool functionality. Consideration should be made to also only de-obfuscating strings in-memory at the moment the data is needed. When a previously de-obfuscated value is no longer needed, it should be wiped from memory. | (S//NF) String data and/or configuration data is very useful to analysts and reverse-engineers. |
| (S//NF) DO NOT decrypt or de-obfuscate all string data or configuration data immediately upon execution. | (S//NF) Raises the difficulty for automated dynamic analysis of the binary to find sensitive data. |

# Development Tradecraft

## Do's

- Obfuscate or encrypt all strings.
- Strip all debug symbols.
- Strip all debugging output.
- Minimize binary file size.
- Use end to end encryption.
- Use RFC compliant network protocols.
- Use GMT/UTC/Zulu as the time zone.
- Use variable size & timing of implant network communications.

## Don'ts

- Have "dirty words" in the binary.
- Perform operation that will cause the target to be unresponsive.
- Leave dates/times.
- Allow network traffic, such as c2. packets to be re-playable.
- Proper cleanup of network connections.
- Assume a "free" PSP product is the same as a "retail" copy.

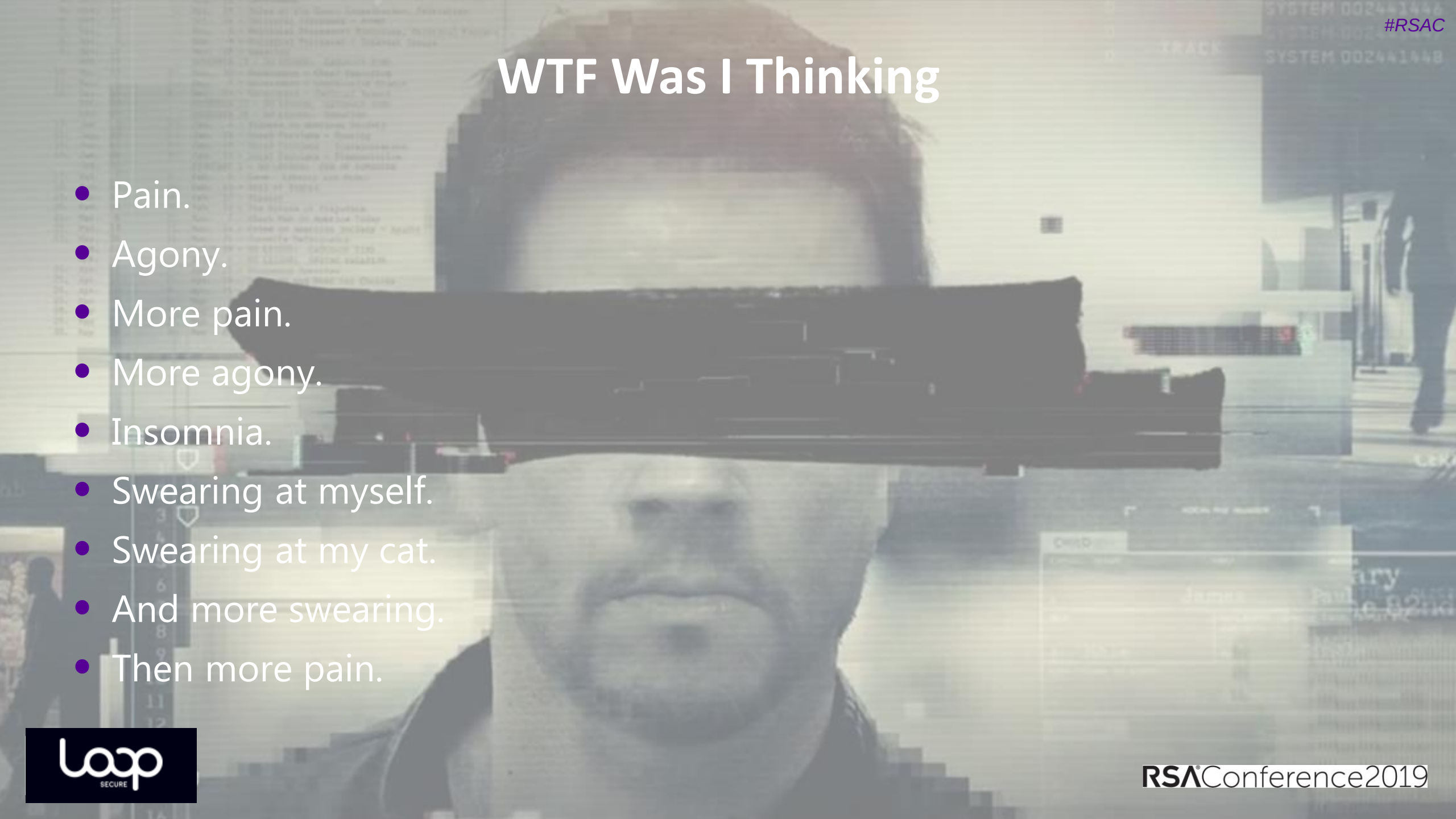
# Keylogger Code

# Keylogger Code Obfuscated

untitled

```
//NUMPAD KEYS
        case 0x60: FileWrite(__ODA__("ERV]]]]]bmw!/==\?IXX]gkmvw#/67\?HIV]bfpy},")); break;
        case 0x61: FileWrite(__ODA__("NSTTTTTT^abfgqv}+466DMY[dsx %03\?HP^ju&2:")); break;
        case 0x62: FileWrite(__ODA__("N\\cmmmmmr||$38ADJKWft|!$'0006\?GR\\is!(,,:")); break;
        case 0x63: FileWrite(__ODA__("CHKQQQQQS^lr /::ELUcly%%3BCIQS[\\^fsuvww{")); break;
        case 0x64: FileWrite(__ODA__("IT[_____jtu&/36>ES`hv%3>MR\\ir!*9EJVadqx}")); break;
        case 0x65: FileWrite(__ODA__("DSSbbbbbktuy(,:HOOSSSYfor}.26=HU_iny*99:")); break;
        case 0x66: FileWrite(__ODA__("BQS`````gnu|\"&034:@IR_nu{*4AMYbegsw%*5DL")); break;
        case 0x67: FileWrite(__ODA__("LX^hhhhhjouvx#))1>ADQT^er}.7DLP_ckov &&4")); break;
        case 0x68: FileWrite(__ODA__("GMQ\\\\\\\\\\emq|&,-17:FILZafors{,/3@BCEER[el")); break;
        case 0x69: FileWrite(__ODA__("GQQ^^^^^aahp},006CITY`mou|\")8\?ABO^mv!&&.")); break
```

# WTF Was I Thinking



- Pain.
- Agony.
- More pain.
- More agony.
- Insomnia.
- Swearing at myself.
- Swearing at my cat.
- And more swearing.
- Then more pain.

```
Done building project "Implant.vcxproj".
========== Build: 3 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

SOMETIMES YOU GOTTA RUN BEFORE YOU CAN WALK

*- Tony Stark / Ironman*

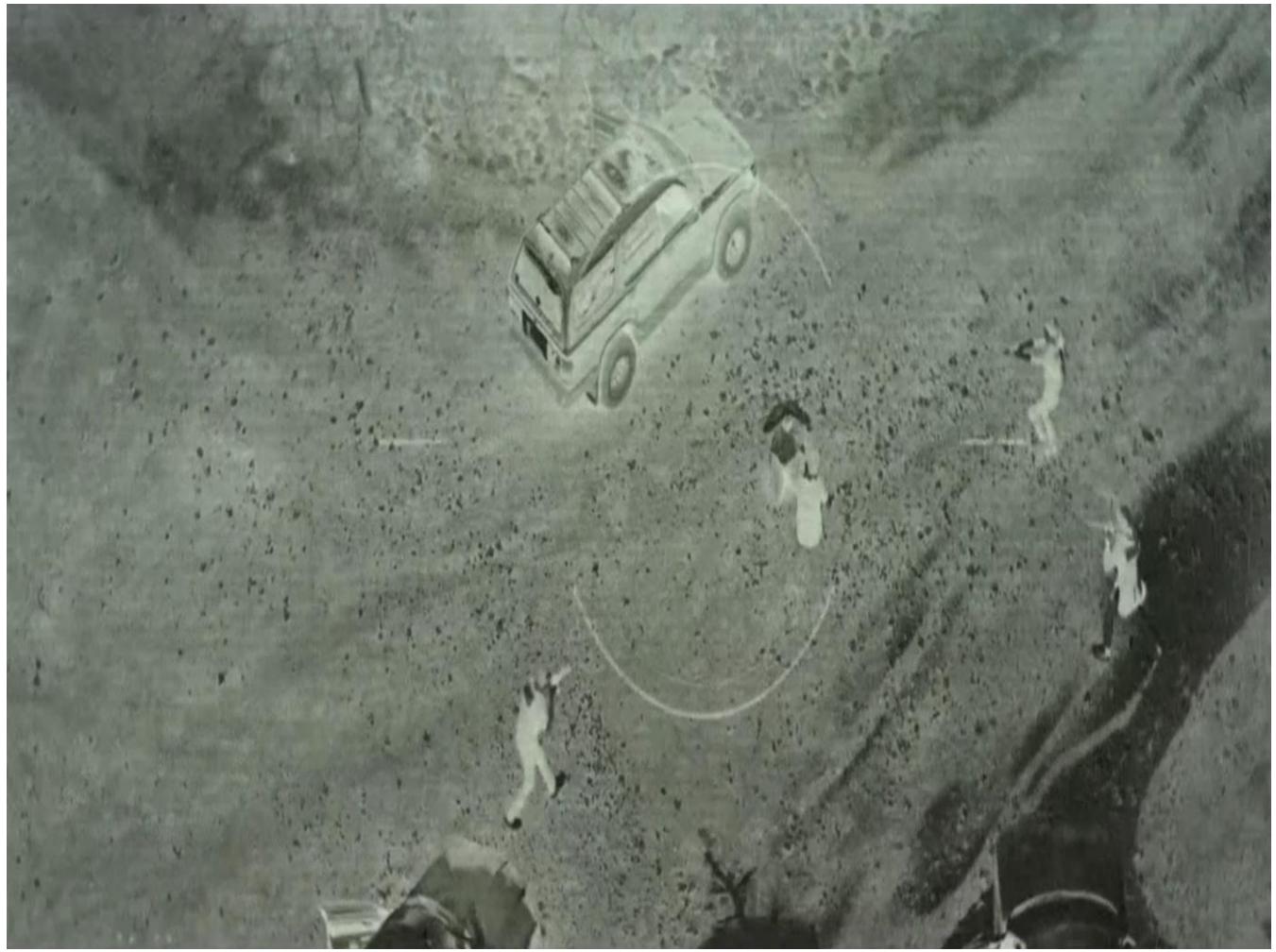
RSAConference2019

# Video

Loop SECURE

# Operational Thoughts

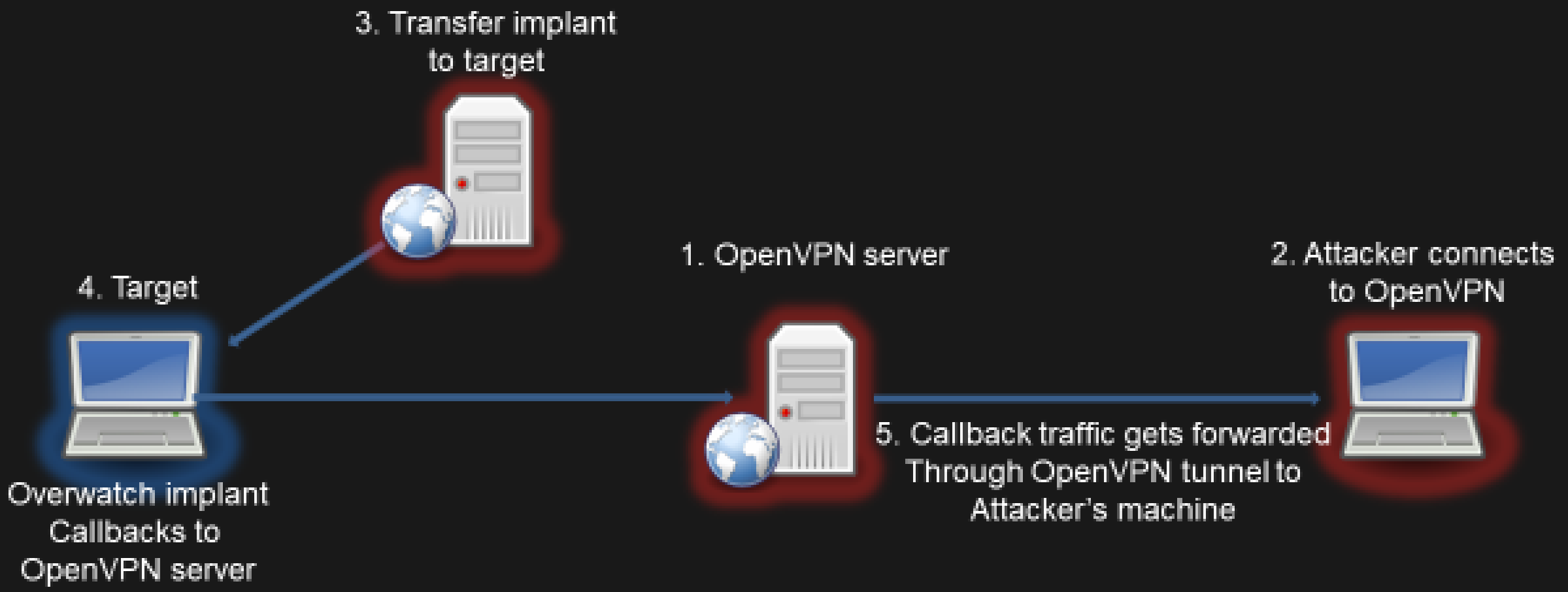
- Not detected.
- Infrastructure tradecraft.
- C2 off the internet.
- EXE needs to blend in.
- Persistence worked.
- Terminal output.
- No PowerShell.
- Need add proxy detection.



# Infrastructure Tradecraft

- Operational security.
- SSL.
- SSH Keys.
- OpenVPN.
- Wiregaurd VPN.
- C2 runs local.
- No C2 on server.

# Infrastructure Deployment



RSAConference2019

# Demo
# Overwatch Offensive

SECRET//ORCON//NOFORN

## 1.1 Concept of Operations

Overwatch Offensive is an automated Implant that provides a simple collection platform on remote computers running the Microsoft Windows operating system. Once the tool is installed on the target, the implant is run within a Windows service process.

Overwatch Offensive will then periodically beacon to its configured listening post(s) to request tasking and deliver results. Communication occurs over one or more transport protocols as configured before or during deployment.

# Why I Did This

- Because it was fun.
- Increase my own skill set.
- To help understand why you should test your security posture against real-world attacks.
- Empower security team to think like an attacker.
- Demonstrate why custom tools can be so devastating to your environment.

RSAConference2019

# Questions