# NECCDC Materials

Ty Tony

# NECCDC 2015

Northeast Collegiate Cyber Defense Competition

# Blue Team Packet

March 20th – 22nd

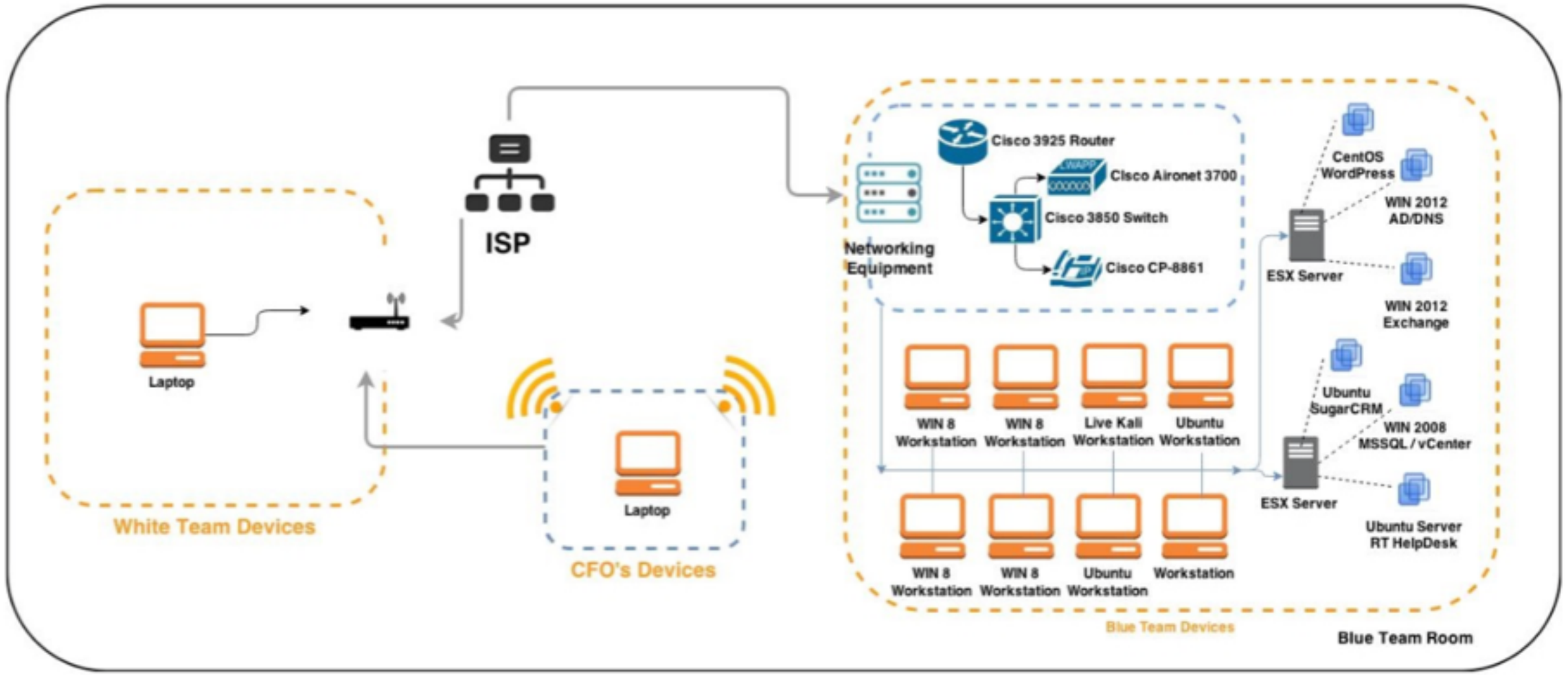
Each team will start the competition with a set of identically configured systems. The objective of the competition is to measure a team's ability to maintain secure computer network operations in a simulated business environment. This is not just a technical competition, but also one built upon the foundation of business operations, policy, and procedures.

A technical success that adversely impacts the business operation will result in a lower score as will a business success which results in security weaknesses. Student teams will be scored on the basis of their ability to detect and respond to outside threats, including cyber-attacks, while maintaining availability of existing network services such as mail servers and web servers, respond to business requests such as the addition or removal of additional services, and balance security against varying business needs.

In an effort to acquire a specific talent pool as well as have success story to showcase their approach, they've acquired a struggling legacy management company named **Enron**. The acquisition has gone smoothly and Skyhook has started to move some of its administrative staff into the new location.

Since Skyhook was only interested in a percentage of current employees, much of the original company's staff was let go or left for "greener" pastures. The cuts and departures included the entire IT staff. Skyhook's CIO, CEO and CFO are on site for the transition, and have brought your team in to evaluate the current state of the IT infrastructure. Have a seat at your new desk, login, and enjoy your new role at Skyhook Incorporated.

# Topology


ISP
Laptop
White Team Devices
Laptop
CFO's Devices
Networking Equipment
Cisco 3925 Router
Cisco Aironet 3700
Cisco 3850 Switch
Cisco CP-8861
CentOS WordPress
WIN 2012 AD/DNS
ESX Server
WIN 2012 Exchange
Ubuntu SugarCRM
WIN 2008 MSSQL / vCenter
ESX Server
Ubuntu Server RT HelpDesk
WIN 8 Workstation
WIN 8 Workstation
Live Kali Workstation
Ubuntu Workstation
WIN 8 Workstation
WIN 8 Workstation
Ubuntu Workstation
Workstation
Blue Team Devices
Blue Team Room

## Initial Services Scored for Enron's Network

| System | IP | Operating System | Scored Services |
|---|---|---|---|
| Vsphere | 10.1.x.210 | ESXi | Web /Uptime |
| Crm | 10.1.x.203 | Linux | Web |
| Website | 10.1.x.212 | Wordpress | Web |
| Ad | 10.1.x.201 | Windows | DNS / AD |
| exchange | 10.1.x.202 | Windows | SMTP / Web |

*** x denotes team number

Each server is critical to the operation of Enron. Note that systems labeled as workstations and laptops must remain end-user systems and cannot be re-provisioned as server systems.

Services may be added to this list or removed from it via injects.

# Scoring Methodology

**Final scores will be awarded using the following point distribution:**

| | |
|---|---|
| **40%** | Functional service uptimes and SLA violations as measured by the scoring engine. |
| **40%** | Successful completion of inject scenarios through the ISE. |
| **20%** | Incident Response and Red Team Activity |

A system restore service is available to teams. This service has a minimum of 15 minutes lead time and could take 30 minutes. There will be a **penalty of 5% per restoration** against the final score for the service(s) restored.

Note: This penalty does not apply to restoration due to hardware failure.

# THE FIRST 10 MINUTES

- You get a blue team packet, prior to hour 0 you know
- Your servers
- Your Services
- Your scored services
- Your networks

# USE YOUR ADVANTAGE

- Change Passwords on Scored Services and Infrastructure first
- Patch after changing passwords if you have to choose between the two
- Configure your firewalls third to prevent egress
- Configure your firewalls to only allow scored services in
- Read up on remote administration of your services

# THE REDTEAM ADVANTAGE

- We only have to find one way in
- You have to find all the ways we can get in

# NATIONAL REDTEAM DAY 1

- Anyone can get in once
- A national redteamer stays in
  - Persistence
  - Agents
  - De-Security
  - Authentication bypass
  - Keylogging
  - Webshells
  - New accounts
  - Stealing all valuable data

# NATIONAL REDTEAM DAY 2

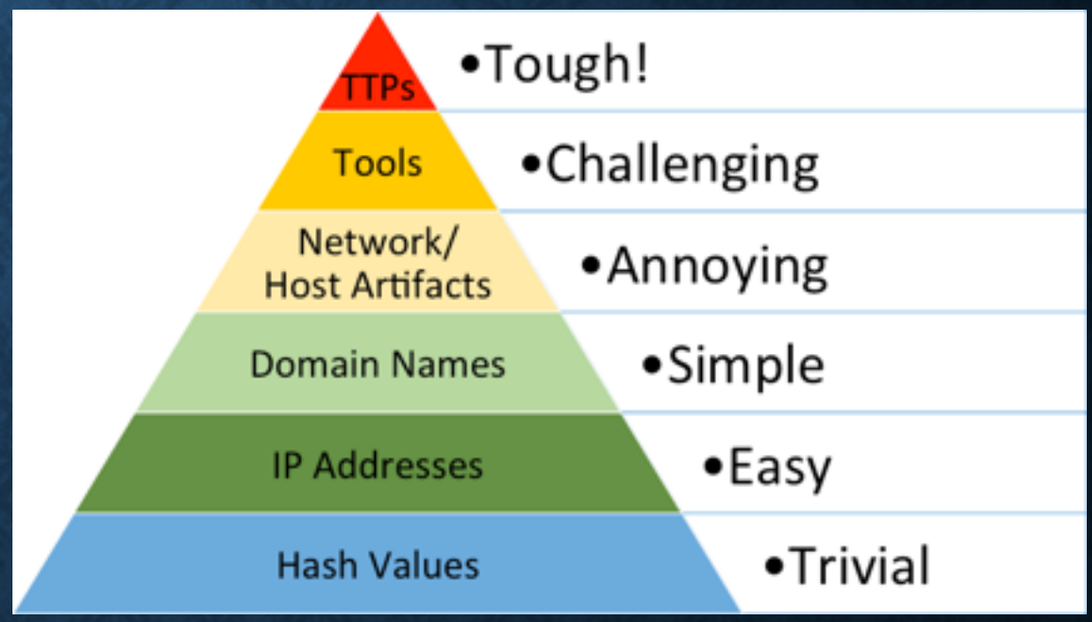
- Exposing PII
- Defacing websites
- Scored service take down
- MBR wiping
- System reposession

# TWO DAYS, TWO LESSONS

- Day 1
  - Learn to Hunt
  - Assume Compromise
- Day 2
  - Respond don't react
  - Root cause analysis

# GOING HUNTING

- Know Normal
  - Network Connections
  - Event Logs/Syslog
  - Weblogs
- Antivirus/Antimalware
- Hash checks
- Use Splunk!



# CONSIDER REAL IR

- Memory Forensics
  - Free and Open Source
    - Rekall – Works on live memory
    - Volatility – Works on memory dumps
  - We can’t hide from memory forensics
- Execution artifacts
- Deleted file recovery

# Forensic Memory Analysis

**We can remember it for you wholesale!**

Michael Cohen
Johannes Stuettgen

# Popular open source tools

- Two popular open source tools:
  - Volatility - Current release 2.3.1 - supports XP-Win7, OSX, and Linux.
    - Supports many Windows versions out of the box with embedded profiles
      - approx 20 different profiles WinXPSP2x86, Win7SP1x64
  - Rekall - A fork (rewrite) of Volatility from 2013.
    - Vastly different design philosophy:
      - Profiles are not distributed with the tool - they are hosted on a public profile repository - Fetched on demand.
      - Approximately 100 different windows kernel versions from WinXP to Win8.1 for x86 and amd64 architectures.

# Rekall vs. Volatility

- Volatility
  - Contains about 20 embedded windows profiles (OSX profiles must be downloaded manually).
  - Requires the user to know which profile to select.
  - Windows Profiles do not contain constants - Most plugins scan/guess offsets of kernel globals.
- Rekall
  - Profile repository contains > 300 profiles, indexed by GUID.
  - Impractical for user to specify (GUID) - profiles are usually autoselected.
  - Profiles contain exact offsets of kernel data

SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

# Rekall Memory Forensic Framework Cheat Sheet v1.2

POCKET REFERENCE GUIDE

by Alissa Torres

## Purpose

The Rekall Memory Forensic Framework has unique syntax and plugin options specific to its features and capabilities. This cheatsheet provides a quick reference for memory analysis operations in Rekall, covering acquisition, live memory analysis and parsing plugins used in the 6-Step Investigative Process. For more information on this tool, visit rekall-forensic.com.

# Rekall Memory Forensic Framework

Memory analysis is one of the most powerful investigation techniques available to forensic examiners. Rekall auto-detects the target system's profile, using a repository of more than 100 kernel versions available either online or stored locally.

When launching Rekall, you can run single commands or drop into an interactive session to take advantage of caching, preventing the need to obtain the same data with subsequent plugin runs.  This cheatsheet shows command line examples using both techniques.

## Windows Memory Acquisition

**Windows WinPmem** *(Open cmd.exe as Administrator)*

*As of winpmem 2.0.1, the default output file format is AFF4*

**Creating an AFF4**

```
C:\> winpmem_<version>.exe -o output.aff4
```

**Extracting the raw memory image from the AFF4**

```
C:\> winpmem<version>.exe output.aff4 --export
PhysicalMemory -o memory.img
```

**Other WinPmem Options:**

```
view aff4 metadata (-V)| elf output (--elf)
```

## Live Windows Memory Analysis

**Windows WinPmem** *(Open cmd.exe as Administrator)*

**Loading the kernel driver on target system**

```
C:\Program Files\Rekall> winpmem<version>.exe -l
```

**Creating Rekall session referencing live memory**

```
C:\Program Files\Rekall> Rekal -f \\.\pmem
[1] pmem 11:14:35> pslist
```

**Creating an image (AFF4) with aff4acquire**

```
[1] pmem 11:14:35> aff4acquire output="w.aff4"
```

# Registry Analysis Plugins

## Enumerate and Extract Registry Hives

**hives**- Find and list available registry hives

```
$ rekal -f image.img hives
```

**regdump**- Extracts target hive

--hive_regex Regex Pattern Matching
- D ”<dir>” Dump directory

```
$ rekal -f image.img regdump --hive_regex="SAM" -D
"/cases"
```

**printkey**- Output a registry key, subkeys, and values

-K “Registry key path”

```
[1] image.img 11:14:35> printkey -K
   “Software\Microsoft\Windows\CurrentVersion\Run”
```

**userassist**- Find and parse userassist key values

## Step 4. Look for Evidence of Code Injection

**malfind** - Find injected code and dump sections

- **pid=<pid>** - Show information only for specific PIDs
- **phys_eprocess=** - Provide physical offset of process to scan
- **eprocess=** - Provide virtual offset for process to scan
- **dump_dir=** - Directory to save memory sections

```
[1] be.aff4 11:14:35> malfind

eprocess=0x853cf460
```

**ldrmodules** - Detect unlinked DLLs

- **verbosity=** - Verbose: show full paths from three DLL lists

```
[1] be.aff4 11:14:35> ldrmodules pid=1936
```

## Step 5. Check for Signs of a Rootkit

**psxview** - Find hidden processes using cross-view

**modscan** - Scan memory for loaded, unloaded, and unlinked drivers

**services** - Enumerates services from in-memory registry hive

**svcscan** - Scans for **_SERVICE_RECORD** records

**hooks_inline** - Detects API hooks

- **eprocess=** Filters by virtual address EProcess
- **phys_eprocess=** Filters by physical address of EProcess

**hooks_eat** - Detects Export Address Table hooks

```
[1] image.img 11:14:35> hooks_eat pid=6764
```

**hooks_iat** - Detects Import Address Table hooks

**ssdt** - Hooks in System Service Descriptor Table

**driverirp** - Identify I/O Request Packet (IRP) hooks

- **regex="drivername"-** Filter on REGEX name pattern

**object_tree** - Tracks named objects

```
[1]image.img 11:15:35> object_tree type_regex="Driver"
```

## Step 6. Dump Suspicious Processes and Drivers

**dump** - Hexdump data starting a specified offset

```
[1] image.img 11:14:35> dump <virtual offset>
```

**dlldump** - Extract DLLs from specific processes

```
[1] image.img 11:14:35> dlldump pid=1004 dump_dir=.
```

**moddump** - Extract kernel drivers

**procinfo** -Dump process to executable sample

```
[1] image.img 11:14:35> dlldump pid=1004 dump_dir=.
```

**procdump** - Dump process to executable sample

**pid=** Dump only specific PIDs

**offset=**Specify process by physical memory offset

**dump-dir=**Directory to save extracted files

```
[1] image.img 11:14:35> procdump proc_regex="csrss"
dump-dir="/tmp"
```

**memdump** - Dump every memory section into a file

(command line options shown below)

**- p <PID>**- Dump memory sections from these PIDs

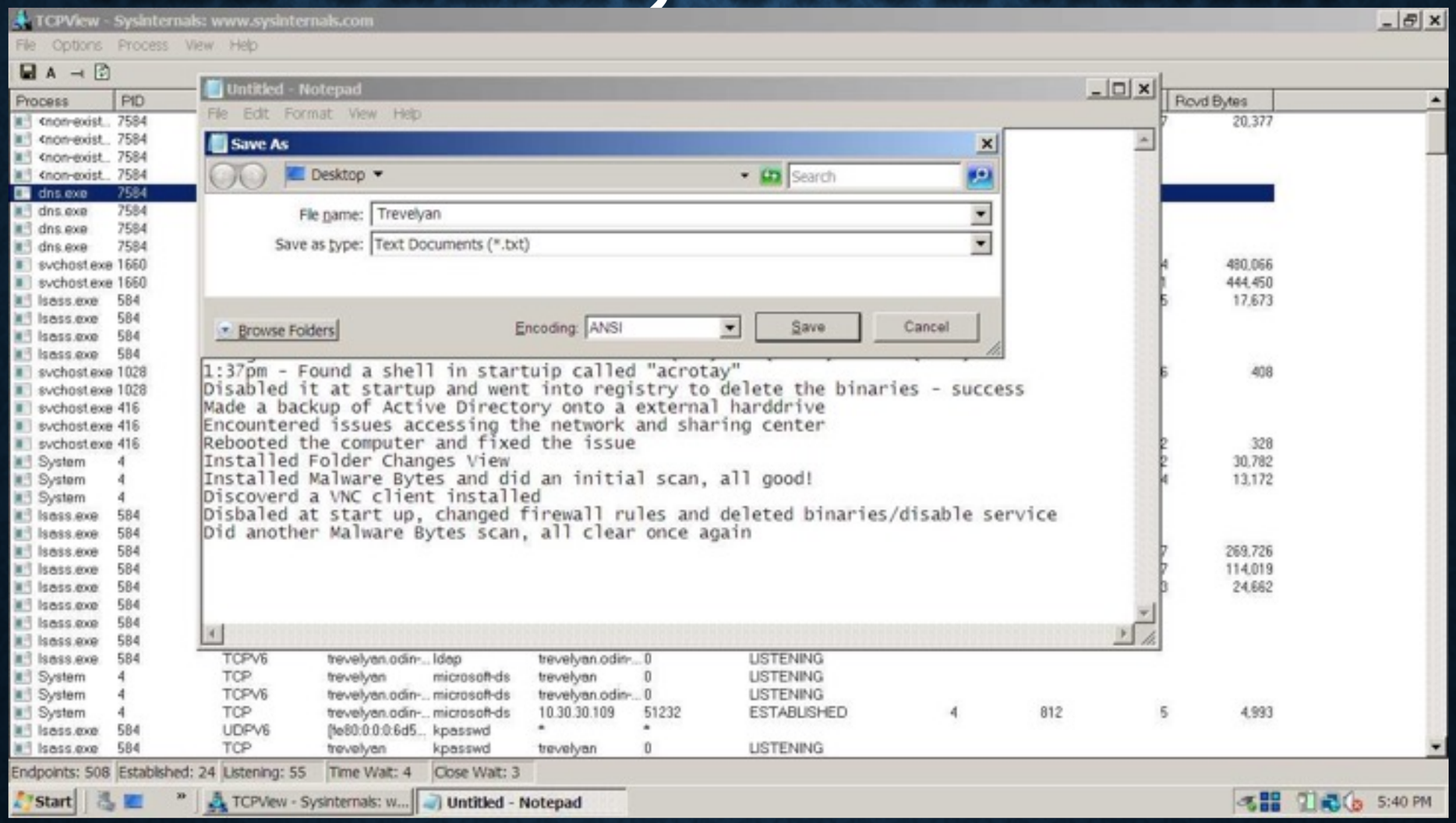
**-D /cases** -Directory to save extracted files

```
# rekal -f image.aff4 memdump -D ./output -p 1004
```

# THINK ABOUT WHY THINGS ARE HAPPENING

- **If a website keeps getting defaced**
- **If a database keeps getting deleted**
- **If a windows domain controller becomes a linux system**
- YOU HAVEN'T FIXED THE PROBLEM

# ALL CLEAR, ONCE AGAIN



# SOMETIMES YOU DO SMART THINGS

```
C:\ProgramData>sb2.exe
This program is blocked by group policy. For more information, contact your system administrator.

C:\ProgramData>
```

# MANY TIMES WE MISS THINGS

- There was a 3rd network?!

# SUMMER SCHOOL FOR BLUE TEAMS

- You need to practice
  - Change all passwords if you've been compromised
  - If you are playing at nationals, you've been compromised
- Learn host firewalls in case you can't get the network firewall working
- Practice learning what a normal system looks like
- Start watching your own network connections
- Practice basic memory forensics