SIU Southern Illinois University
CARBONDALE

Department of Computer Science
Southern Illinois University Carbondale

CS 491/531
SECURITY IN CYBER-PHYSICAL SYSTEMS

Lecture 13: Risk and Vulnerability in ICS

DR. ABDULLAH AYDEGER

LOCATION: ENGINEERING A 409F

EMAIL: AYDEGER@CS.SIU.EDU

Outline

Risks

Security Tests

System Characterization

Recall: Lessons learned from Stuxnet

Previous Belief	Lesson Learned
Control systems can be isolated from other networks, eliminate risk of cyber incident	They are still subject to human who can use USB
PLC and RTUs don't run modern OS, don't have necessary attack surface	PLCs can be affected and have been affected by malware
Firewall/IDS are sufficient	Blacklisting based defense is not sufficient due to zero-day vulnerabilities, whitelist defenses should be considered against unknown exploits

Recall: How to proceed if infection detected

Not to clean it directly

- May have subsequent levels of infection that exist (staying idle and undetected)
- Valuable info such as infection path, other compromised hosts

First step to isolate the infected host

Collect as much as possible forensics data

- System logs, network traffic, memory analysis data

Statistics of ICS Incidents

80% impacting ICS are “unintentional”

- Only 35% from outsider
- Insider + unintentional is a big concern

Embedded devices and network appliances were targeted 34%

- Windows-based ICS and enterprise hosts 66%

These numbers would help to understand risks that should be prioritized

https://scadahacker.com/

What is Risk?

ISO defines: “potential that a given threat will exploit vulnerabilities of asset”

Risk is a function of:

- The likelihood of a given “Threat Event”
- Exercising particular potential vulnerability of an asset
- Consequences that impact operation of the asset

Threat Event:

- Threat source and actor to carry out the event
- Threat vector to initiate the event
- Threat target which the event attacks

Risk Cont'd

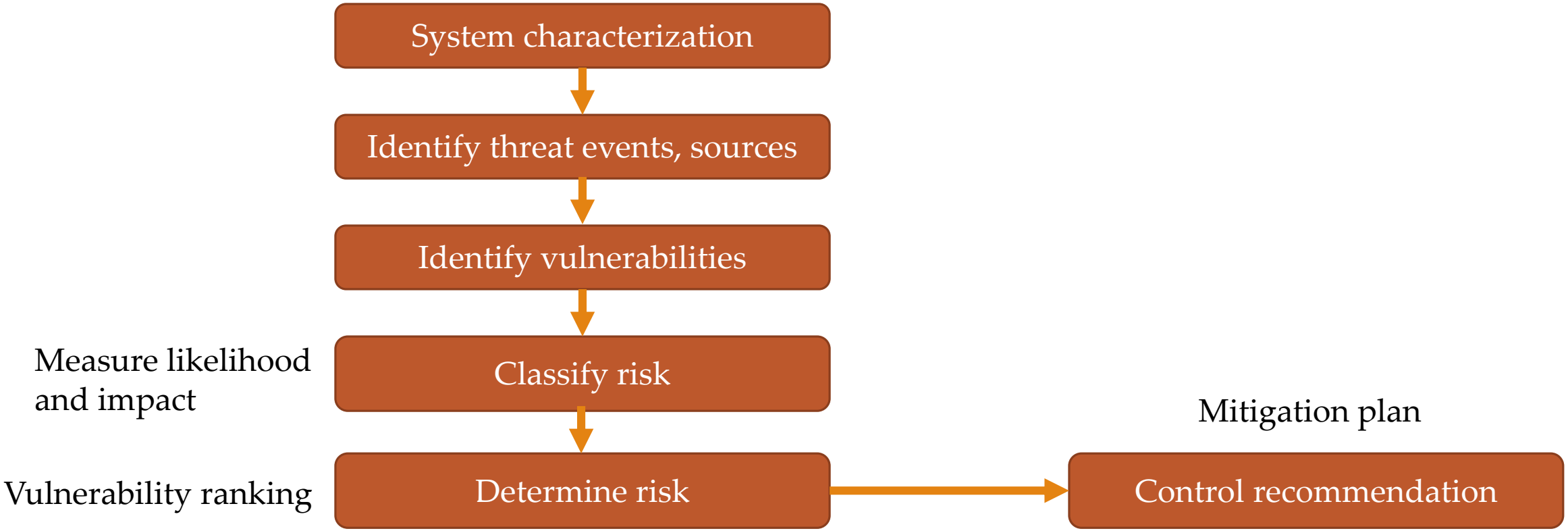
Threat source and actor refer to human aspect of attack:

- Capability to carry out the attack
- Intent to cause harm
- Opportunity to initiate the event

Insider has extensive Capability and Opportunity

- Thus, very likely to be target in early phases of blended attack due to being harder to detect and mitigate

Flowchart of Assessing Risks to ICS



Security Tests

Benefit = number of vulnerabilities identified

Any assessment is a snapshot in time

- Vulnerabilities are discovered, disclosed, and patched within short time (daily, weekly)
- Requires repetitive process

Repetition can be triggered in:

- Changes to system such as component upgrade
- Changes in threats such as new exploit

Objective of Security Testing

Identification of system vulnerabilities

Detection of security controls employed

- And their effectiveness

https://tools.kali.org/stress-testing/termineter

https://github.com/inguardians/optiguard

Security Testing in ICS

Penetration testing in ICS?

- Requires non-production test environment

Security Audits

- Test particular system against specific set of policies, procedures or regulations
  - It usually mean known threats
  - Do not uncover unexpected or latest vulnerabilities

Security and Vulnerability Assessment

- To look at the entire solution for the system
  - This means each ICS system and subsystem/network infrastructure and so on

Theoretical Tests

Industrial systems operational integrity is critical to allow test to be run, even small risk tests can disrupt the integrity (time requirements, etc.)

- Leads to theoretical tests

Standardized method of completing questionnaire

- Like interview

Dept of Homeland Sec (DHS) ICS Cyber Emergency Response Team (ICS-CERT) developed Cyber Security Evaluation tool (CSET) for offline tests

- Security practices are compared against recognized industry standards
- Answers generate output with the recommendation list

Online – Offline Physical Tests

Online test:

- Evaluation is performed on actual running industrial network
  - Contains volatile ICS components
- Represents completely functional and operational ICS architecture
  - Including 3rd party components

Offline test:

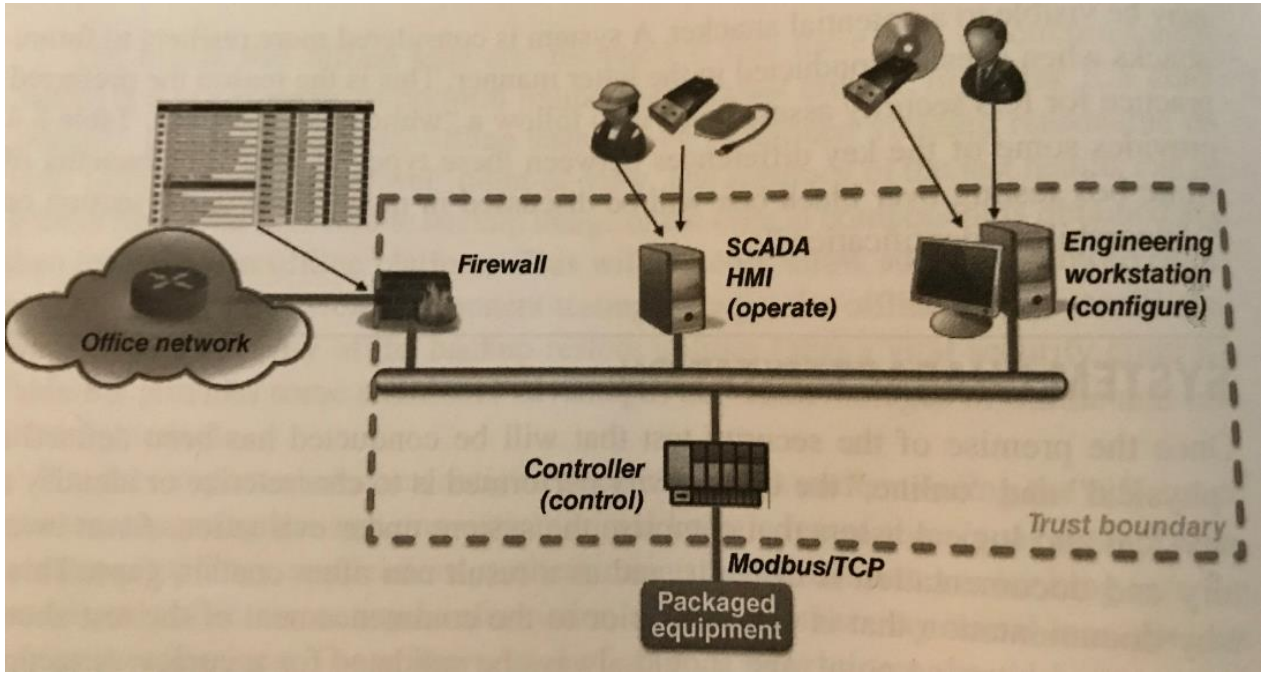
- Not connected to physical process and not performing real-time control operations
  - Difficult to include 3rd party
- Reflects subset of overall architecture, can omit key components

System Characterization

First activity to perform for physical and online test

Use zone concept for better analysis

- Create trust boundary
- All external entry points require penetration



Identifying Entry Points

Entry Point	Description	Data Flows associated	Assets associated
Firewall	Internal firewall between office and control network	Authentication File Sharing	Engineering Workstation (EWS)
Modbus port on Controller		Modbus/tcp	Controller
USB Port	On EWS	Software, Data files	EWS
Wireless	WLAN/Bluetooth on EWS	Software, Data files	EWS

Identifying Logical Assets

Physical Asset	Logical Asset	Threat Event
Firewall	Firmware Management Port/Config ID & Authentication Services Log Files Communication interfaces	Modify firmware to change behavior Modify configuration Elevation of privilege Modify Logs to remove audit DoS
Network	Switch Ports Switch config	Malicious connection Modify behavior
Controller	Modbus/Ethernet interface	Send elicit instructions (malware), DoS
EWS	Windows Stored files Configuration Ethernet interface/modem Keyboard/CD/USB	Elevation of privilege Copy/modify/delete information Modify config, send command to controller Gain Remote access, inject malware Modify anything, inject malware

Scanning Industrial Networks

Device Scanners

Vulnerability Scanners

Traffic Scanner

Device Scanners

Ping command:

- Basic device identification tool, built-in to most commercial OS
- Not effective in ICS due to security appliances rarely forward ping (ICMP)

Arping and arp-scan:

- Based on ARP protocol (MAC layer) can be used to identify hosts

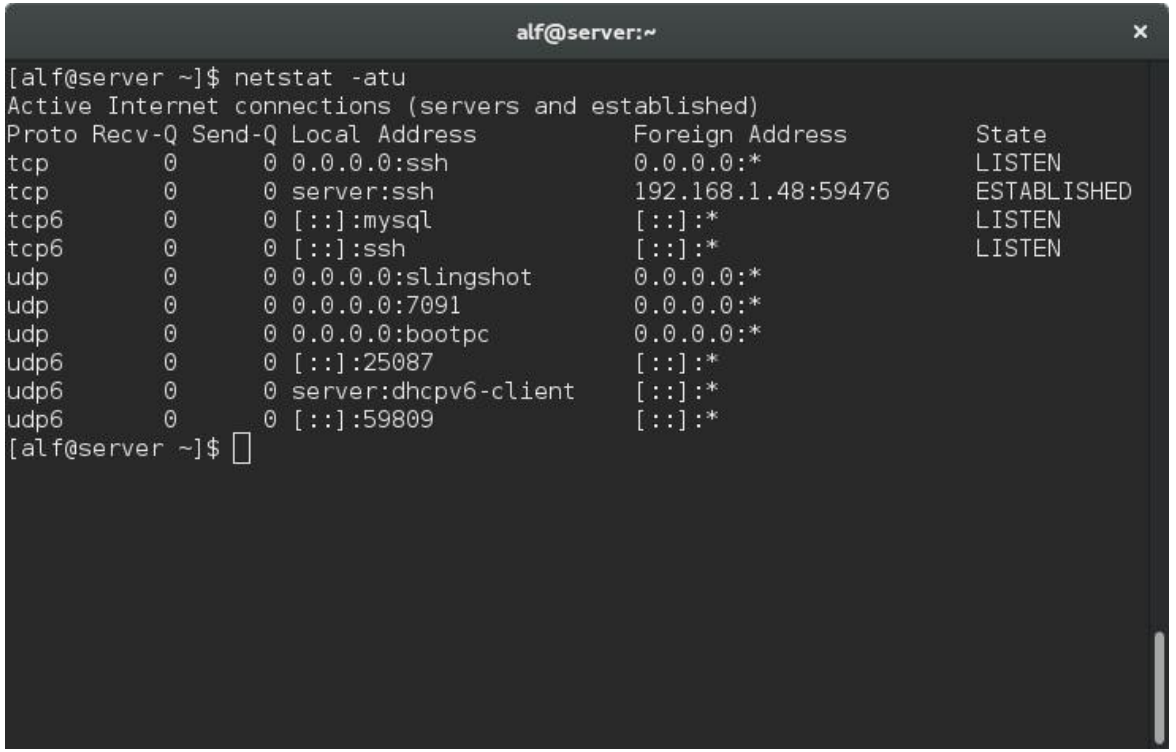
Network mapper or *nmap*:

- Data collection via network-based, external packet injection and analysis
- Host discovery, host service detection, OS detection, spoofing, execute customized code

Device Scanners

Network statistics or netstat tool

- Command-line feature is available on most OS
- Useful when trying to identify applications and services running on particular host
- Does not inject packets on network which could compromise time-sensitive communication between ICS
- Friendly and passive



Vulnerability Scanners

OpenVAS open-source, and many commercial tools (Tenable Nessus, SAINT scanner)

Identify vulnerabilities that may exists comparing with database of known vulnerabilities

- Depends on product's database, different results

https://tools.kali.org/vulnerability-analysis/openvas

Traffic Scanners

Collect raw network packets and provide them for host identification, firewall rule set, etc.

Basic form is tcpdump for Linux, windump for Windows

- Purpose is to capture and save network traffic

Wireshark

- Uses pcap (file style of tcpdump)
- Used for analysis of network traffic
- Not recommended to use for raw packet collection
  - Memory performance issues

Wireshark INP Dissectors

CIP

EtherCAT

Ethernet POWERLINK

GOOSE

Modbus

OPC UA

PROFINET

SERCOS

Examples of Live Host Identification

Quiet Scanning Techniques:

- Single ARP request via arping

```
root@debian:~# arping -c 2 192.168.178.27
ARPING 192.168.178.27
60 bytes from 08:00:27:c9:7c:85 (192.168.178.27): index=0 time=396.617 usec
60 bytes from 08:00:27:c9:7c:85 (192.168.178.27): index=1 time=313.585 usec

--- 192.168.178.27 statistics ---
2 packets transmitted, 2 packets received,   0% unanswered (0 extra)
rtt min/avg/max/std-dev = 0.314/0.355/0.397/0.042 ms
root@debian:~#
```

- Scan entire subnet via arp-scan (-l)

```
File  Edit  View  Search  Terminal  Help
root@kali:~# arp-scan --interface=eth0 --localnet
Interface: eth0, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9 with 256 hosts (http://www.nta-monitor.com/tools/arp-scan/)
172.16.44.1     00:50:56:c0:00:08       VMware, Inc.
172.16.44.2     00:50:56:fa:49:a4       VMware, Inc.
172.16.44.140   00:0c:29:2d:9c:10       VMware, Inc.
172.16.44.141   00:0c:29:0a:56:4f       VMware, Inc.
172.16.44.145   00:0c:29:5f:1d:1f       VMware, Inc.
172.16.44.148   00:0c:29:0f:46:91       VMware, Inc.
172.16.44.149   00:0c:29:df:37:17       VMware, Inc.
172.16.44.153   00:0c:29:ec:fd:52       VMware, Inc.
172.16.44.254   00:50:56:fe:c9:1a       VMware, Inc.

9 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9: 256 hosts scanned in 2.203 seconds (116.21 hosts/sec). 9 responded
root@kali:~#
```

Examples of Live Host Identification

- Packet capture without attempting to resolve addresses to hostname via tcpdump

```
TCPDUMP(8)                        System Manager's Manual                        TCPDUMP(8)

NAME
       tcpdump - dump traffic on a network

SYNOPSIS
       tcpdump [ -AbdDefhHIJKlLnNOpqRStuUvxX# ] [ -B buffer_size ]
               [ -c count ]
               [ -C file_size ] [ -G rotate_seconds ] [ -F file ]
               [ -i interface ] [ -j tstamp_type ] [ -m module ] [ -M secret ]
               [ --number ] [ -Q in|out|inout ]
               [ -r file ] [ -V file ] [ -s snaplen ] [ -T type ] [ -w file ]
               [ -W filecount ]
               [ -E spi@ipaddr algo:secret,...  ]
               [ -y datalinktype ] [ -z postrotate-command ] [ -Z user ]
               [ --time-stamp-precision=tstamp_precision ]
               [ --immediate-mode ] [ --version ]
               [ expression ]

DESCRIPTION
       Tcpdump  prints  out  a  description of the contents of packets on a network interface
       that match the boolean expression; the  description  is  preceded  by  a  time  stamp,
       printed,  by default, as hours, minutes, seconds, and fractions of a second since mid-
       night.  It can also be run with the -w flag, which causes it to save the  packet  data
       to  a file for later analysis, and/or with the -r flag, which causes it to read from a
       saved packet file rather than to read packets from a network  interface  (please  note
       tcpdump  is  protected via an enforcing apparmor(7) profile in Ubuntu which limits the
       files tcpdump may access).  It can also be run with the -V flag, which  causes  it  to
       read  a  list  of saved packet files. In all cases, only packets that match expression
       will be processed by tcpdump.
```

Examples of Live Host Identification

Noisy/Dangerous Scanning Techniques:

- Ping sweep on a single subnet via nmap:

```
root@Qhacker:~# nmap -sn 192.168.56.0/24

Starting Nmap 6.46 ( http://nmap.org ) at 2014-06-19 07:38 IST
Nmap scan report for 192.168.56.100
Host is up (0.00058s latency).
MAC Address: 08:00:27:7A:CC:DB (Cadmus Computer Systems)
Nmap scan report for 192.168.56.103
Host is up (0.0017s latency).
MAC Address: 08:00:27:FC:15:EA (Cadmus Computer Systems)
Nmap scan report for 192.168.56.110
Host is up (0.00023s latency).
MAC Address: 08:00:27:00:24:06 (Cadmus Computer Systems)
Nmap scan report for 192.168.56.115
Host is up (0.011s latency).
MAC Address: 08:00:27:A0:16:85 (Cadmus Computer Systems)
Nmap scan report for 192.168.56.113
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 28.97 seconds
root@Qhacker:~#
```

- Create and send specific packets on network via hping3

```
root@ddos: ~
File  Edit  View  Search  Terminal  Help
root@ddos:~# hping3 -h
usage: hping3 host [options]
  -h  --help      show this help
  -v  --version   show version
  -c  --count     packet count
  -i  --interval  wait (uX for X microseconds, for example -i u1000)
      --fast      alias for -i u10000 (10 packets for second)
      --faster    alias for -i u1000 (100 packets for second)
      --flood      sent packets as fast as possible. Don't show replies.
  -n  --numeric   numeric output
  -q  --quiet     quiet
  -I  --interface interface name (otherwise default routing interface)
  -V  --verbose   verbose mode
  -D  --debug     debugging info
  -z  --bind      bind ctrl+z to ttl           (default to dst port)
  -Z  --unbind    unbind ctrl+z
      --beep      beep for every matching packet received
Mode
  default mode     TCP
  -0  --rawip      RAW IP mode
  -1  --icmp       ICMP mode
  -2  --udp        UDP mode
  -8  --scan       SCAN mode.
                   Example: hping --scan 1-30,70-90 -S www.target.host
```

Suggested ICS Actions

Instead of ping sweep:

- Perform physical verification
- Conduct passive network listening
- Scan subset of targets

Instead of port scan:

- Do local verification (netstat)
- Scan duplicate or test system on non-production network

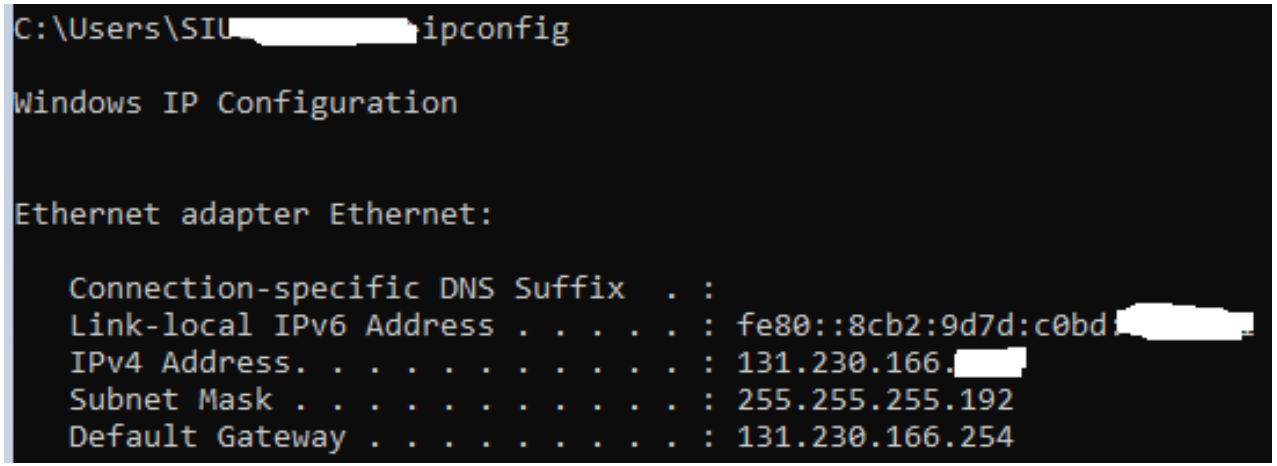
Instead of vulnerability scan:

- Non-production network

Command Line Tools

No packet injection

To display network configuration values, *ipconfig* can be used



Command Line Tools

To determine what apps running and how they map to communication ports via netstat

```
C:\Users\sarav>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    10.211.55.3:52992      40.90.189.152:https    ESTABLISHED
  TCP    10.211.55.3:53030      13.107.42.254:https    TIME_WAIT
  TCP    10.211.55.3:53031      13.107.18.254:https    TIME_WAIT
  TCP    10.211.55.3:53032      131.253.33.254:https   TIME_WAIT
  TCP    10.211.55.3:53033      204.79.197.222:https   ESTABLISHED
  TCP    10.211.55.3:53034      a-0001:https           ESTABLISHED
  TCP    10.211.55.3:53035      13.107.18.11:https     ESTABLISHED
  TCP    10.211.55.3:53036      13.107.246.254:https   ESTABLISHED
  TCP    10.211.55.3:53037      117.18.232.200:https   ESTABLISHED
  TCP    10.211.55.3:53040      13.107.246.10:https    ESTABLISHED
  TCP    10.211.55.3:53041      ec2-54-167-36-150:ms-wbt-server  ESTABLISHED
  TCP    127.0.0.1:7778         mylocalserver:51929    ESTABLISHED
  TCP    127.0.0.1:49681        mylocalserver:49682    ESTABLISHED
  TCP    127.0.0.1:49682        mylocalserver:49681    ESTABLISHED
  TCP    127.0.0.1:51927        mylocalserver:51928    ESTABLISHED
  TCP    127.0.0.1:51928        mylocalserver:51927    ESTABLISHED
  TCP    127.0.0.1:51929        mylocalserver:7778     ESTABLISHED
  TCP    127.0.0.1:51930        mylocalserver:51931    ESTABLISHED
  TCP    127.0.0.1:51931        mylocalserver:51930    ESTABLISHED
  TCP    127.0.0.1:51932        mylocalserver:51933    ESTABLISHED
  TCP    127.0.0.1:51933        mylocalserver:51932    ESTABLISHED
```

Command Line Tools

A few other examples:

- To provide list of running applications and services with their associated PID via “tasklist”
- To see hardware and software inventory (configs), “systeminfo”
- Window Management Instrumentation Command-line (wmic) provides set of system management features

Steps to be taken for System Characterization

Use arp-scan to identify network-connected hosts

Confirm identified hosts are authorized for the network. If not, physically inspect and take actions. Update system architecture with newly discovered info

Collect host info for each connected device, including hardware and OS info

- Can be obtained via systeminfo

Collect app info for each device including vendor, name, patches, etc.

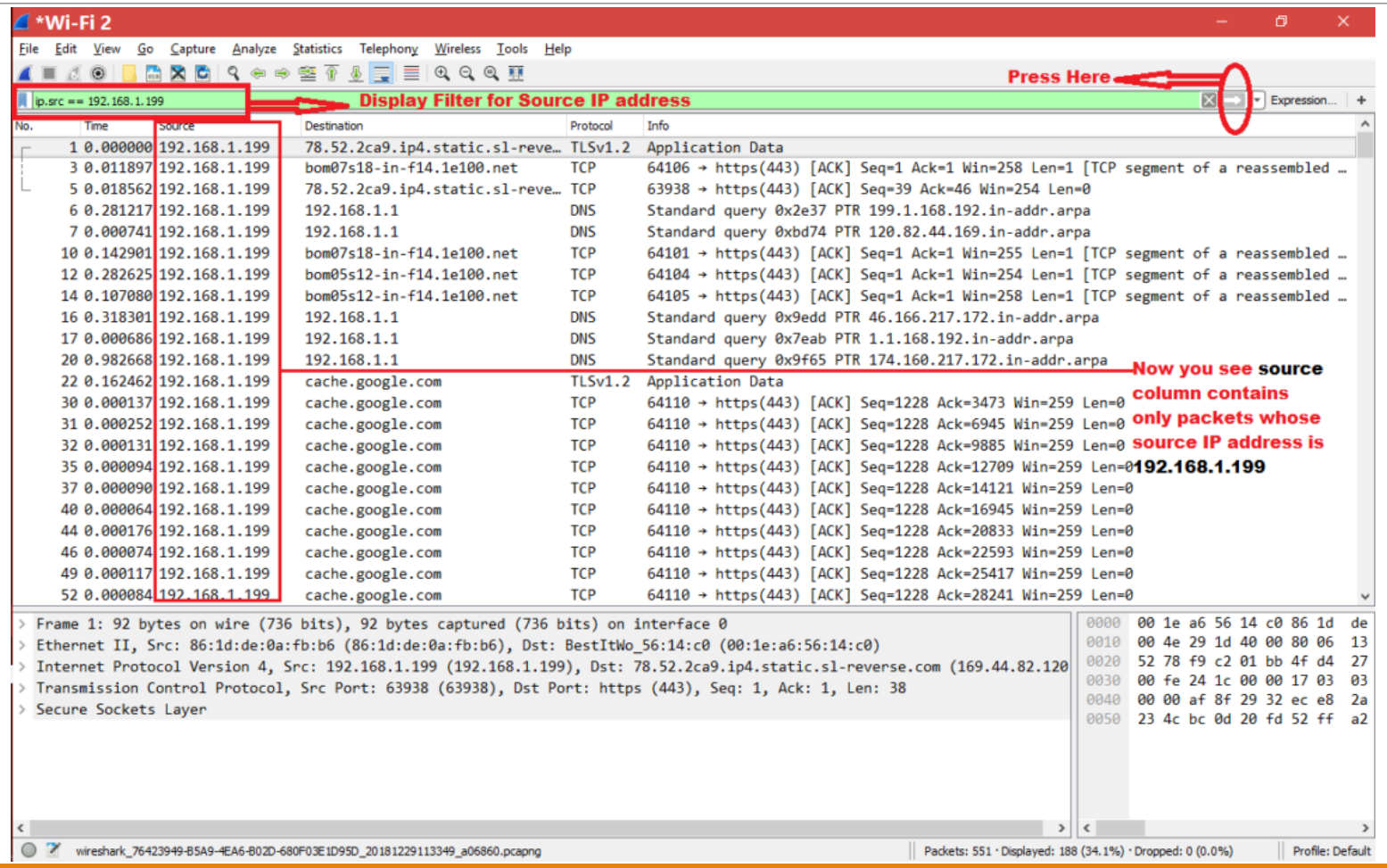
- Can be obtained via wmic

Consolidate this info into database with appropriate classified policies

Data Flow Analysis

Wireshark

Example



Identification of Threats during Security Assessment

Threats could be revealed in following cases (not limited to):

- Infected media discovered from antivirus log
- Corrupted data discovered from local disk evaluation
- Data stolen discovered from network resource usage

Useful info to be used for action plan and mitigation controls