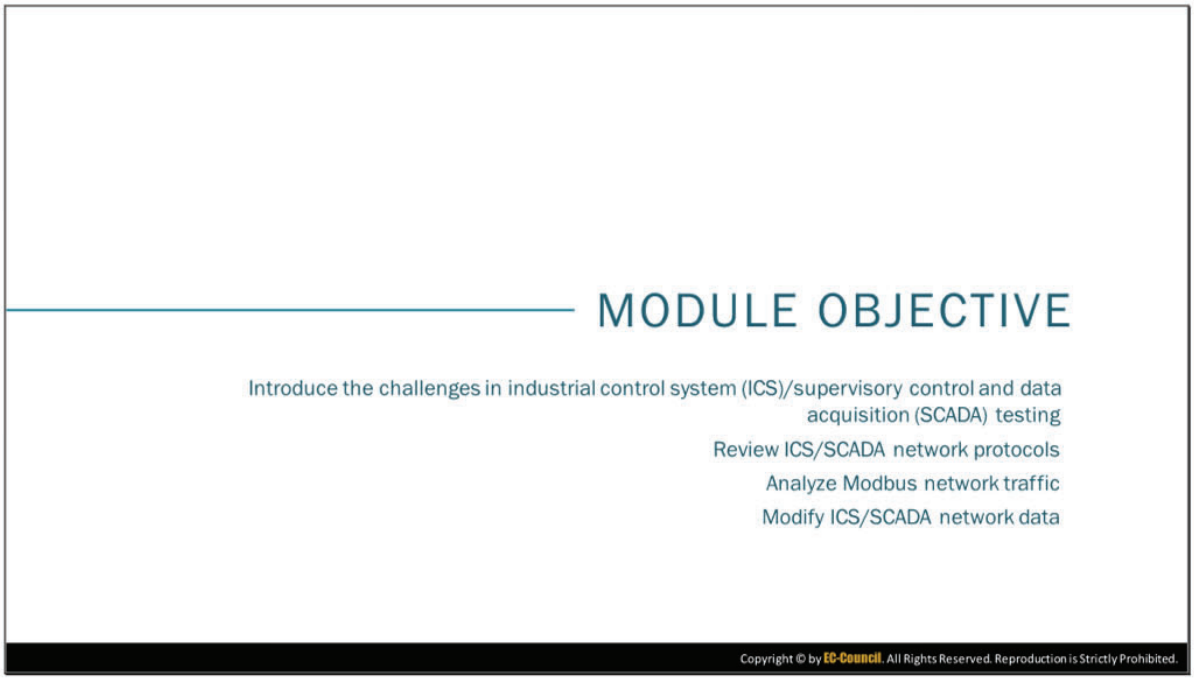
C|PENT

Certified Penetration Testing Professional

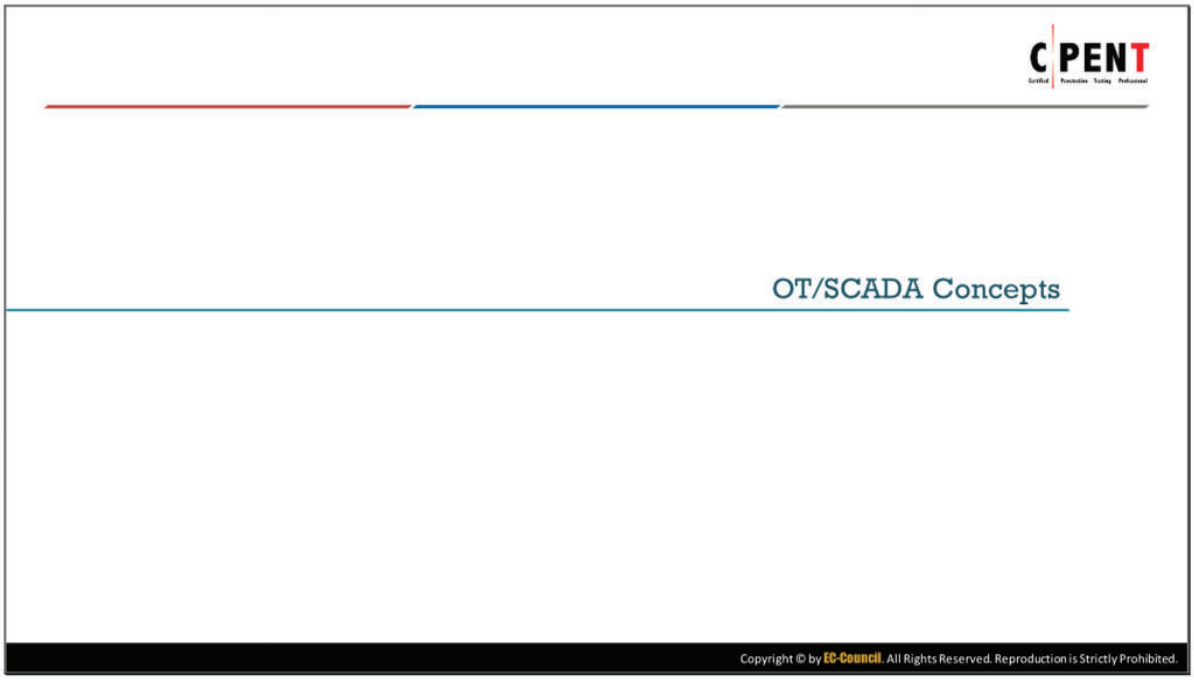
Module 11

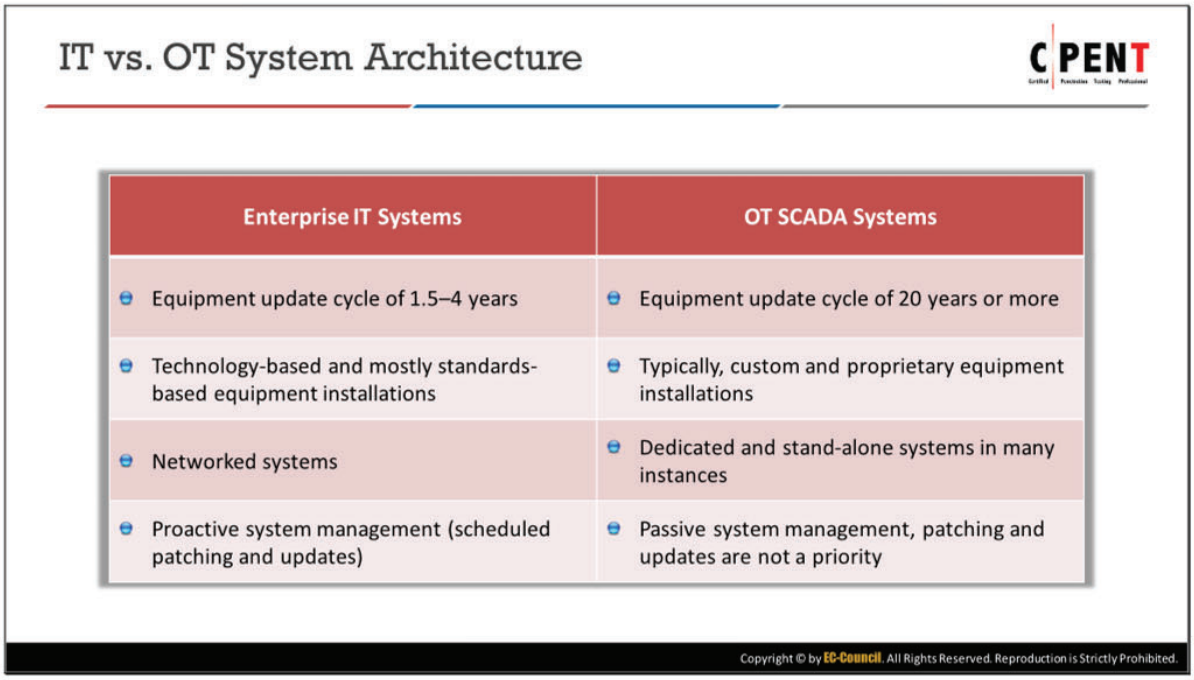
OT and SCADA Penetration Testing



Module Objective

- Introduce the challenges in industrial control system (ICS)/supervisory control and data acquisition (SCADA) testing
- Review ICS/SCADA network protocols
- Analyze Modbus network traffic
- Modify ICS/SCADA network data





OT/SCADA Concepts

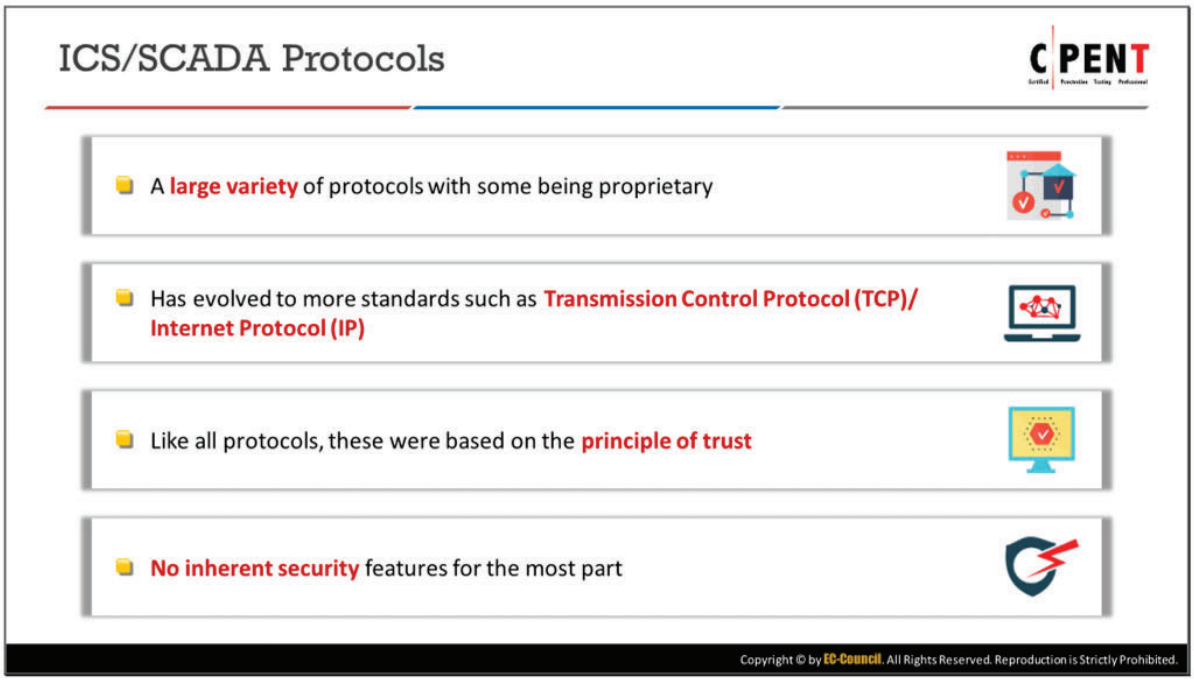
IT vs. OT System Architecture

The types of networks discovered in operational technology (OT)/supervisory control and data acquisition (SCADA) architectures have several differences. One of the most significant is the lifetime of the systems; within critical infrastructure, systems can and often do remain in place for many years.

Another aspect to consider is that some of these systems operate in or around harsh environments. In fact, it is considered normal for SCADA environments to consist of extreme temperatures, dirt, fumes, radiation, and so on.

Enterprise IT Systems	OT SCADA Systems
▪ Equipment update cycle of 1.5–4 years	▪ Equipment update cycle of 20 years or more
▪ Technology-based and mostly standards-based equipment installations	▪ Typically, custom and proprietary equipment installations
▪ Proactive system management (scheduled patching and updates)	▪ Passive system management, patching and updates are not a priority

Table 11.1: IT vs. OT system architecture

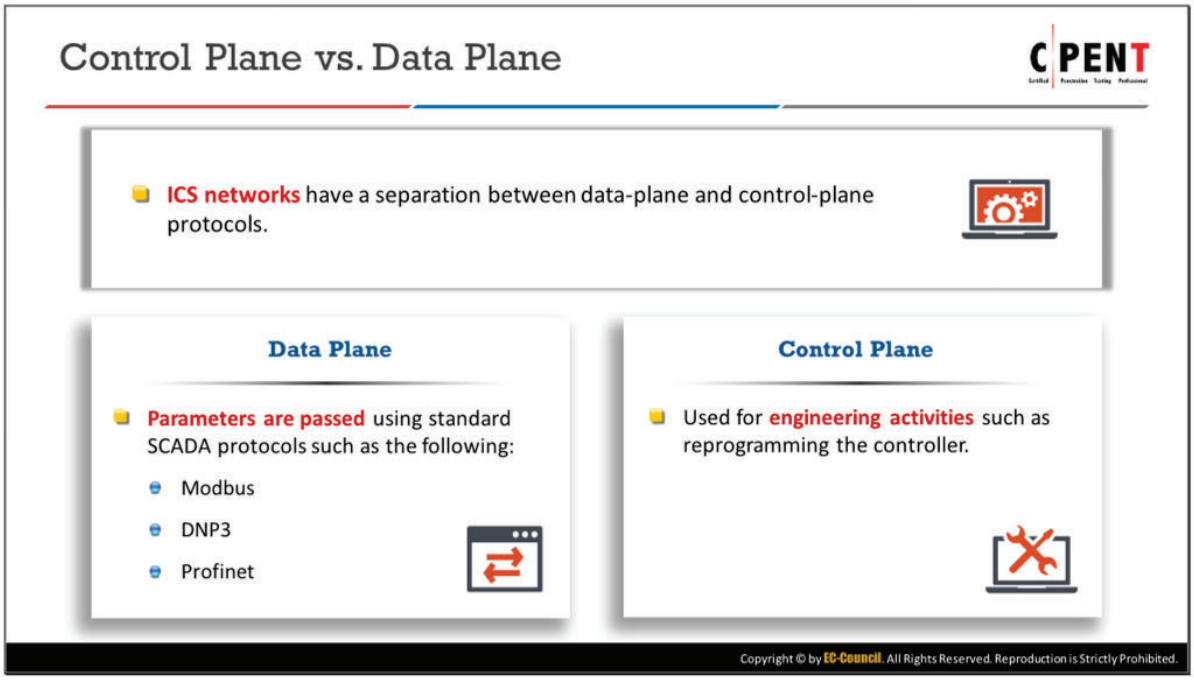


ICS/SCADA Protocols

Industrial protocols have evolved over time from simplistic networks that collect information from remote devices to complicated networks of systems that have redundancy built into the devices. Protocols utilized inside industrial control systems are occasionally very specific to the application.

Over the years, many efforts have been made by standards organizations such as IEC, ISO, and ANSI to standardize protocols. There is an abundance of proprietary protocols as well. In most cases, these protocols are built by vendors to require specific software and hardware to create vendor-centric systems. Irrespective of the origin of the protocol, most industrial protocols have one aspect in common: they were not designed with security in mind and are inherently insecure.

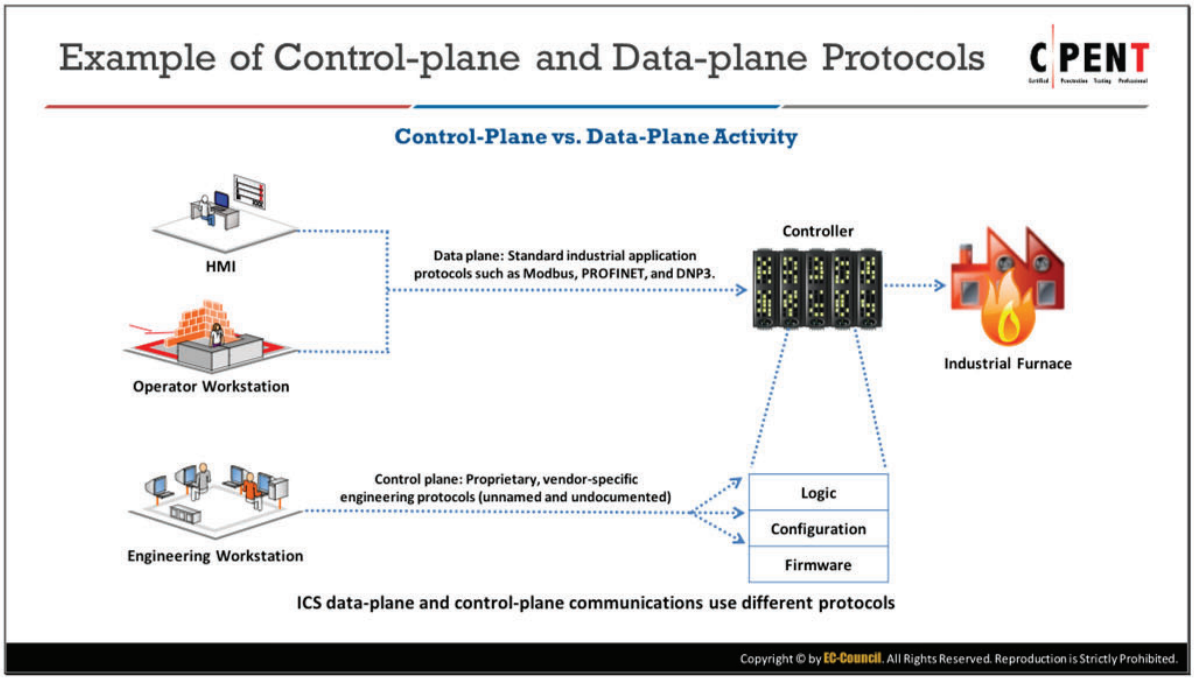
This has become a significant problem since their convergence with IT networks and the now-dominant Transmission Control Protocol (TCP)/Internet Protocol (IP)-based protocols. Understanding these security flaws and how they are exploited is crucial for penetration testing and threat modeling in industrial control system (ICS) environments.



Control Plane vs. Data Plane

Unlike IT networks, ICS environments have a separation between data-plane and control-plane protocols.

- **Data-plane protocols:** These protocols are used to pass parameters and registers between human–machine interface (HMI) SCADA applications and I/O. Standard HMI and SCADA protocols such as Modbus, Profinet, and DNP3 are used here and are fully documented because vendors allow integration with all types of software solutions.
- **Control-plane protocols:** These protocols Include all engineering activities, i.e., reprogramming of the controller, upload/download of the firmware, etc. These are unique engineering protocols because automation vendors created specific software to develop them. The reasoning was that only vendor software should be used to design these devices. Therefore, in addition to being proprietary, these protocols are usually undocumented and unnamed. This makes it difficult to monitor them. Moreover, they are difficult to test because there are few tools to do so. Since these are largely proprietary protocols, it is difficult to write tools that can test all different factors. Furthermore, some of these protocols were created for a specific task and have no documentation, and unless someone took the time to name and define a protocol, they are mostly unnamed.



Example of Control-plane and Data-plane Protocols

HMI
Operator Workstation
Data plane: Standard industrial application protocols such as Modbus, PROFINET, and DNP3.
Controller
Industrial Furnace
Engineering Workstation
Control plane: Proprietary, vendor-specific engineering protocols (unnamed and undocumented)
Logic
Configuration
Firmware
ICS data-plane and control-plane communications use different protocols

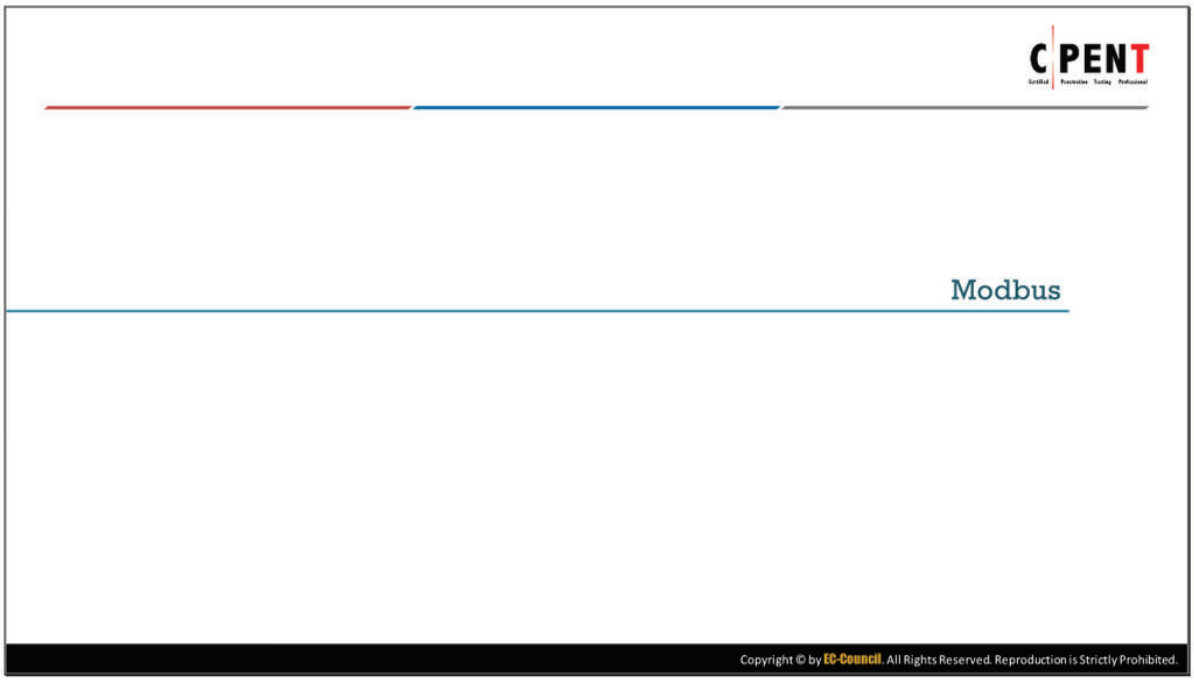
Figure 11.1: Illustration of control-plane and data-plane protocols

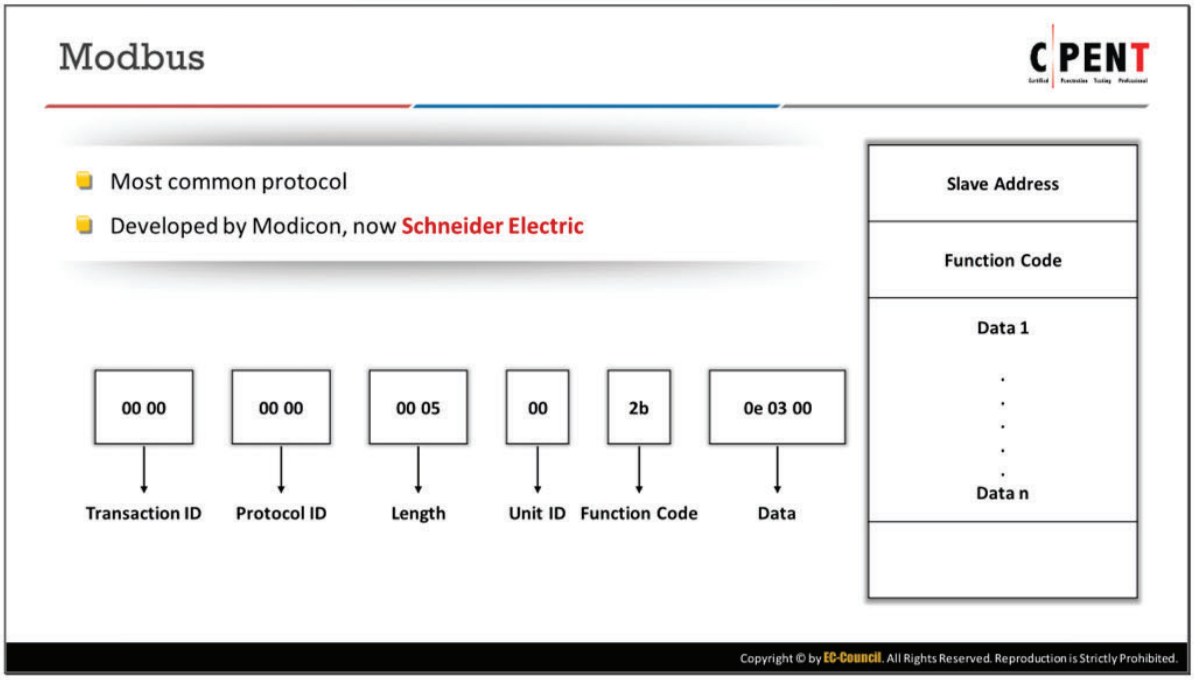
As shown in the figure, penetration testers face challenges when testing in the control-plane side of a network. A common type of vulnerability is that in the programmable logic controller (PLC). An example is CVE-2017-16740. The vulnerability referenced here is within the control plane.

A description is as follows:

A buffer overflow issue was discovered in Rockwell Automation Allen-Bradley MicroLogix 1400 Controllers, Series B and C Versions 21.002 and earlier. The stack-based buffer overflow vulnerability has been identified, which may allow remote code execution.

This is an example of what a penetration tester much research if they encounter these protocols in testing. Since this is a Rockwell controller, there should be a good amount of documentation on it. As described above, there is a classic stack-based buffer overflow vulnerability. Since programmers are usually creatures of habit, more such vulnerabilities are likely to exist.





Modbus

SCADA Modbus is an application-layer messaging protocol positioned at level 7 of the Open Systems Interconnection (OSI) model. It provides client/server communication between devices connected to different types of buses or networks. Developed and published by Modicon in 1979 for use with its PLCs, Modbus is a method used for transmitting information over serial lines between electronic devices. The device requesting the information is called the Modbus master,

and the devices supplying information are Modbus slaves. In a standard Modbus network, there is one master and up to 247 slaves, each with a unique slave Address from 1 to 247. The master can also write information to the slaves.

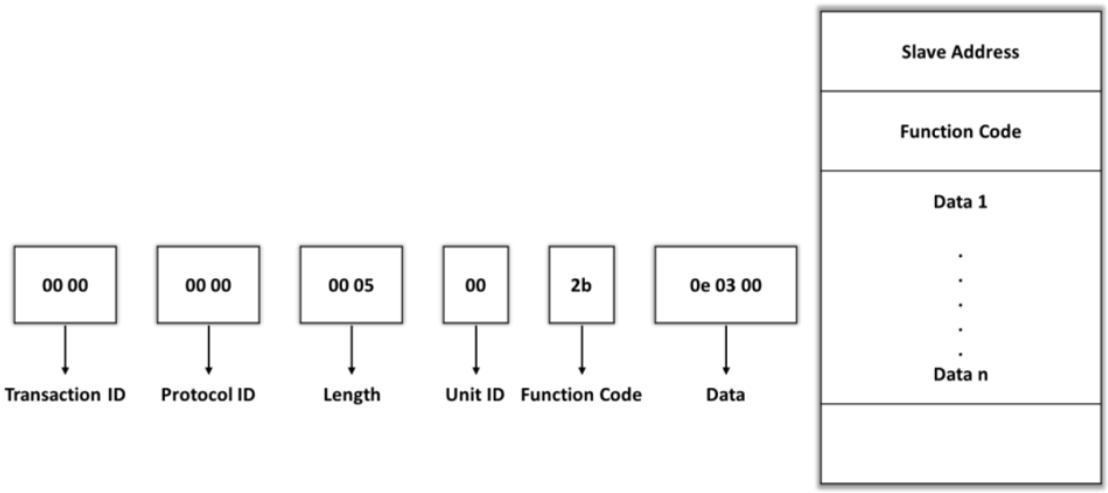
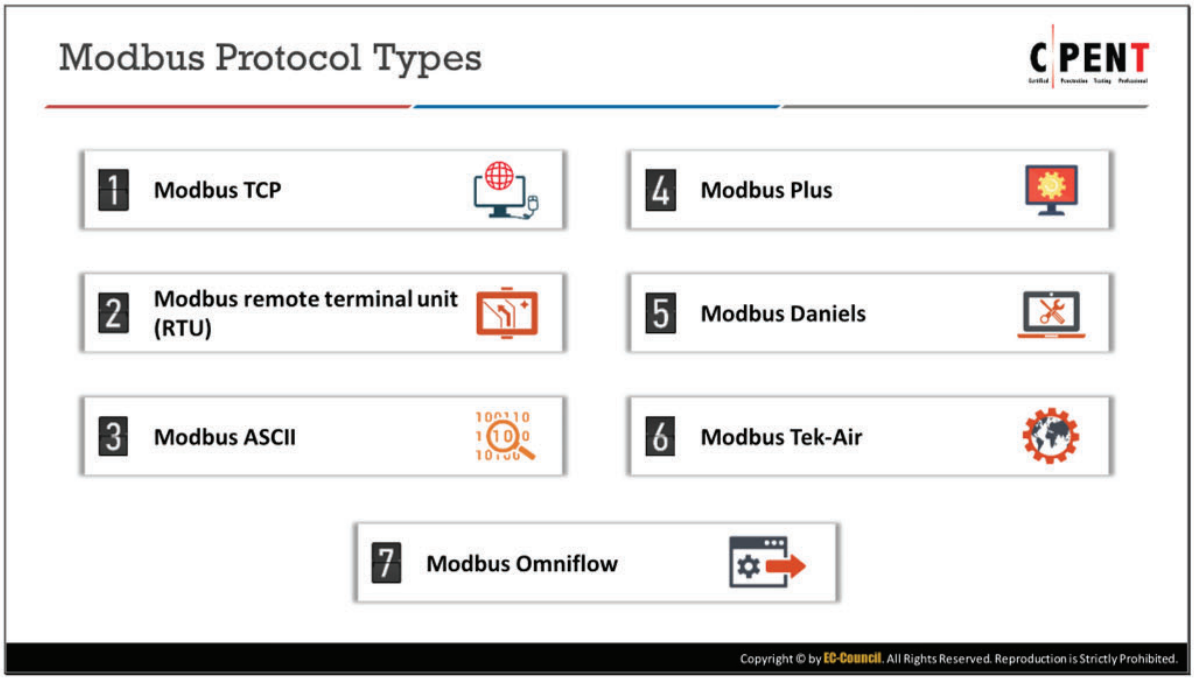


Figure 11.2: Modbus message structure

The protocol's simplicity and efficiency caused it to become the most widely used network protocol in the industrial manufacturing environment. It has been implemented by hundreds of vendors on thousands of different devices to transfer discrete/analog I/O and register data between control devices.

The Modbus protocol has been the most common SCADA protocol. The main reasons for the use of Modbus in an industrial environment are as follows:

- It has been developed with industrial applications in mind.
- It is openly published and royalty-free.
- It is easy to deploy and maintain.
- It moves raw bits or words without placing many restrictions on vendors.
- SCADA systems using Modbus protocols are readily available.



Modbus Protocol Types

The most popular Modbus protocols are Modbus remote terminal unit (RTU) and Modbus TCP/IP.

- **Modbus RTU** is a serial communication protocol that connects different devices on the same network, enabling communication between them.
- **Modbus TCP/IP** covers the use of Modbus communication via an "Intranet" or "Internet" environment using TCP/IP protocols. The most common use of the protocols is currently for the Ethernet attachment of PLCs, I/O modules, and gateways to other simple field buses or I/O networks.

There will always be the question of why the connection-oriented TCP/IP protocol is used rather than the datagram-oriented User Datagram Protocol (UDP). The primary reason is to keep control of an individual "communication" by isolating it in a connection that can be identified, cancelled, and supervised without the need for specific actions on client and server applications. This provides the mechanism a tolerance to network performance changes as well as the possibility to add security features such as firewalls and proxies.

Modbus TCP/IP handles two different situations. A connection can be recognized too easily at the protocol level. A single connection can be used to perform multiple independent communications. In addition, TCP/IP allows a huge number of concurrent connections in case the user decides to re-use an old connection or reconnect to a frequently used connection.

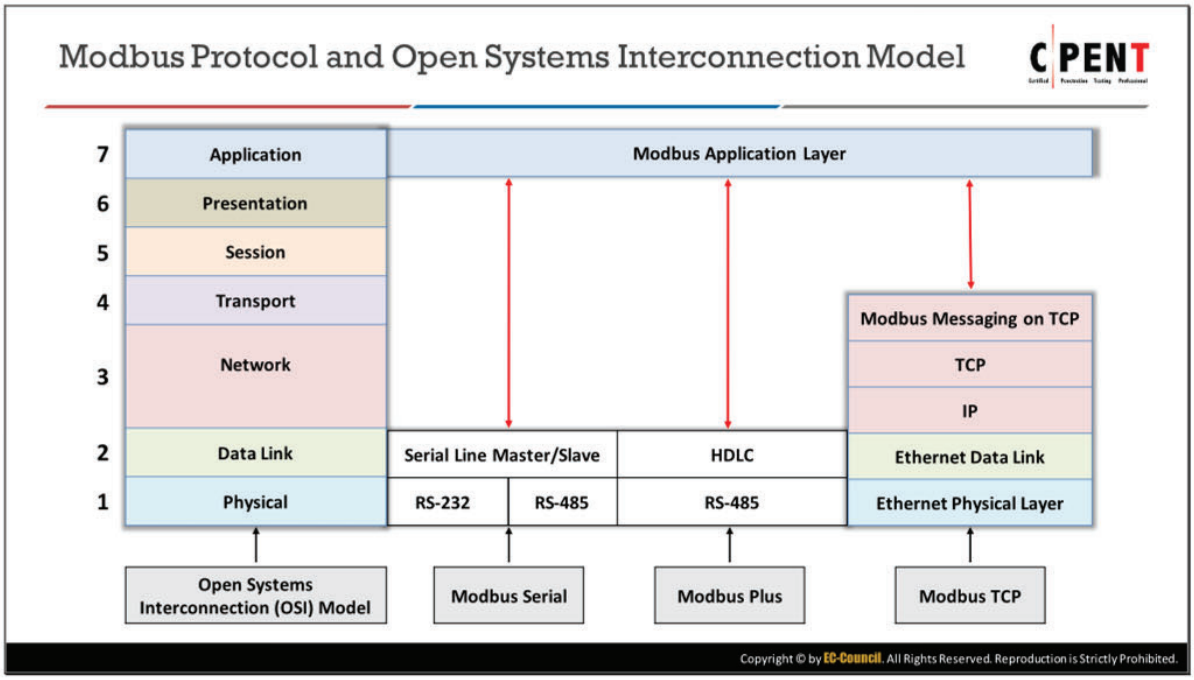
How does Modbus TCP/IP work?

The Modbus device can be connected using an Ethernet port on a gateway. A user can make a query using any standard Modbus Scanner to extract the value from a Modbus device. All requests are sent via TCP/IP on the registered port 502.

What is the difference between Modbus RTU, TCP/IP, and ASCII?

The Modbus protocol defines a protocol data unit (PDU) that is independent of the underlying communication layers. Modbus RTU is the most commonly used and is a binary representation of the PDU with addressing before the PDU and a cyclic redundancy check (CRC) appended after the PDU. Modbus ASCII is a representation of the same PDU using all printable characters (and generally twice as many bytes). Modbus TCP/IP is essentially the same PDU as Modbus RTU, except that the CRC is left out of the application-layer byte string and included in the TCP layer to be dealt with automatically. Furthermore, there are some additional addressing bytes in the TCP encapsulation of the RTU packet.

The function codes, register numbering, and addressing are identical for all protocol variants. Register types are the same, i.e., the same data blocks are defined for a Modbus device.



Modbus Protocol and Open Systems Interconnection Model

Data are stored in Modbus as follows:

- Information in the slave device is stored in four different tables.
- Each table has 9999 data points that can store different values.
- Each coil has 1 bit and is given a data address between x0000 and x9999.
- Each register is 1 word = 16 bits = 2 bytes and has a data address between x0000 and x9999.

The numbers 40001, 10000, etc. should be taken as the address of the location. These numbers are never displayed in actual messages. The actual address is the offset at the location where the number is stored. Therefore, if the device has application-defined units (ADUs), the address 0001 in Modbus and PDU would be at offset 40001 and 40002, respectively.

Each device in a network must be assigned a unique unit address, which can be between 1 and 255. When the master requests for data, the first byte it sends is the device/slave address. Therefore, the slave decides to response or ignore only after the first byte is sent.

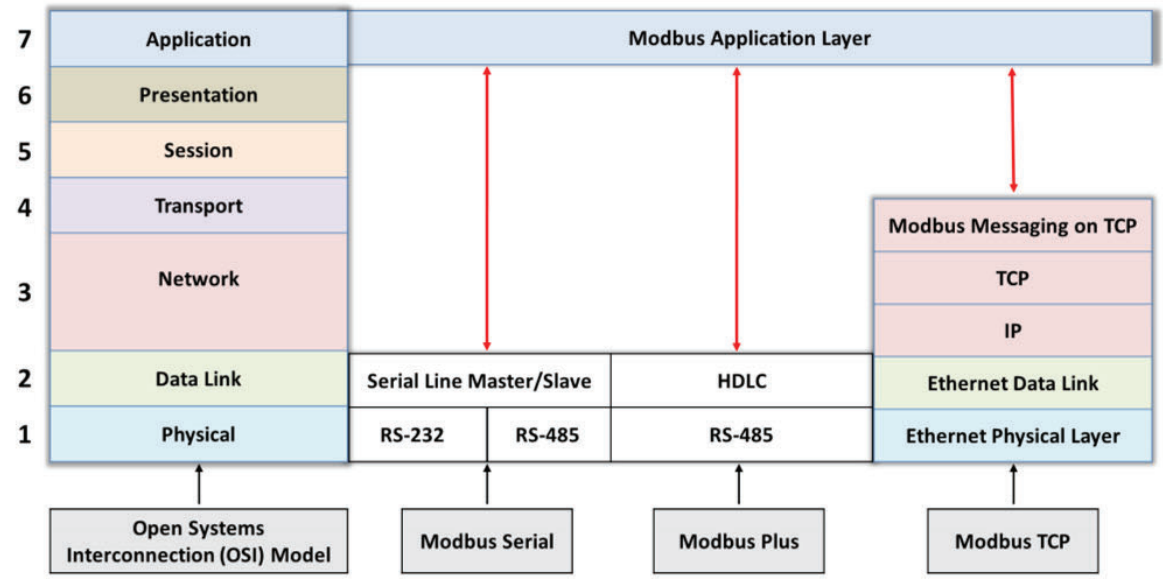
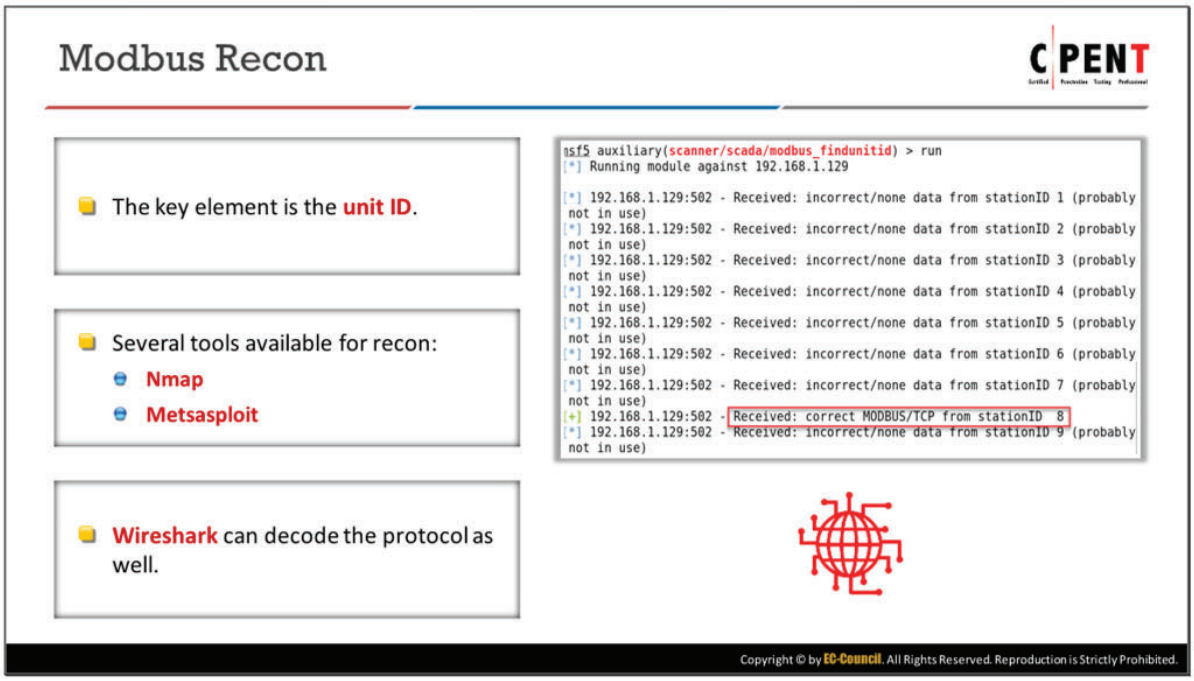


Figure 11.3: Modbus protocol vs. open systems interconnection model

A Modbus map is a simple list of points for any device that supports Modbus as a communication protocol. The Modbus map should have the following information:

1. The kind of data the device reads and stores (e.g., temperature or pressure)
2. The address at which the data are stored (e.g., voltage A-N at 40001)
3. The format in which the data is stored (e.g., bits, UINT16, and SINT16)
4. Engineering units of the points, if required
5. Whether the device has ADU or PDU addressing

The popular network scanning tool Nmap has a script in its repository that can be used to detect and extract data from the Modbus protocol.



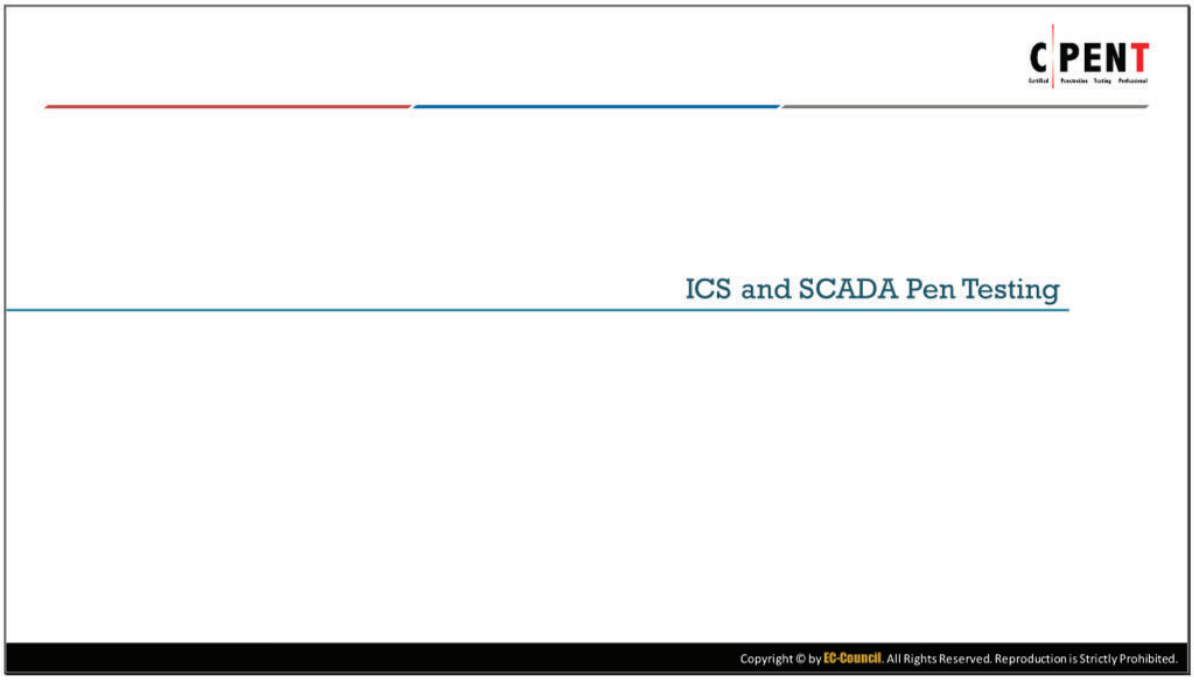
Modbus Recon

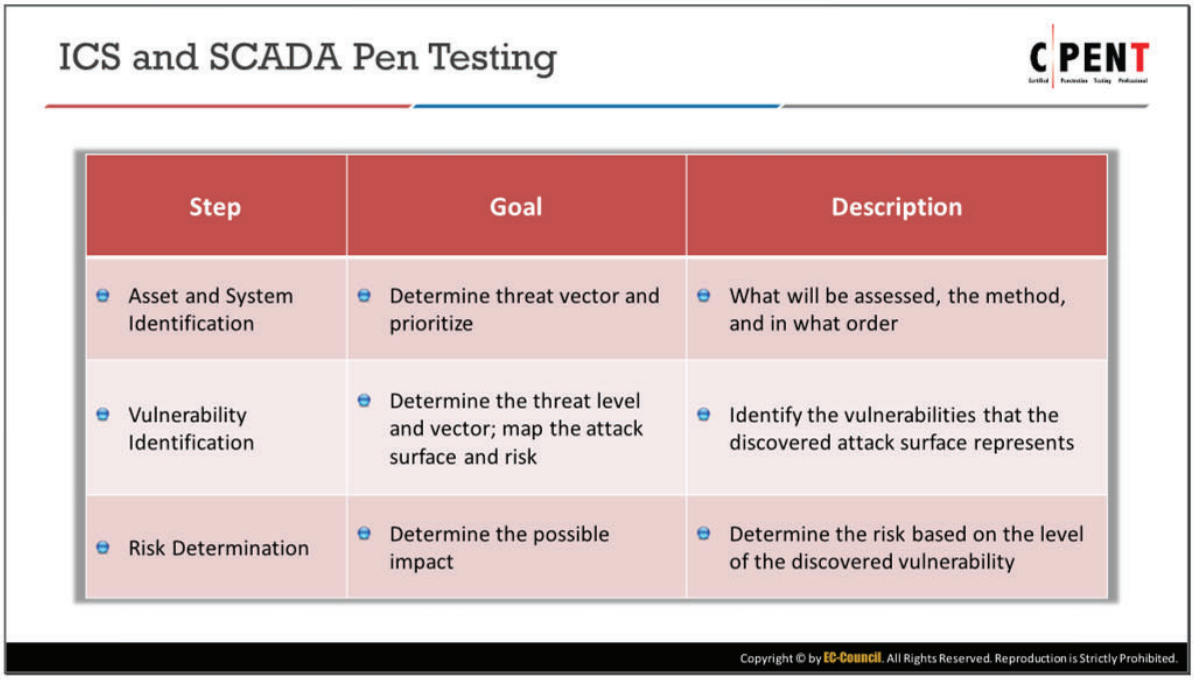
The Internet community can access Modbus at the reserved system port 502 on the TCP/IP stack. A simple request-reply scheme is used for all Modbus application protocol (MBAP) transactions. The client device (also known as master device) initiates a request, and the server (also known as slave) replies. For example, when an HMI workstation requires a value from a PLC, it sends a request message to start the data transfer process. The PLC then sends a response providing the requested information. In this situation, the device running the HMI acts as the client/master, and the PLC acts as the server/slave. Each message contains a function code set by the client/master and indicates to the server/slave the kind of action to perform. Function codes are numbers that tell the slave which table to access and whether to read from or write to the table.

```
msf5 auxiliary(scanner/scada/modbus_findunitid) > run
[*] Running module against 192.168.1.129

[*] 192.168.1.129:502 - Received: incorrect/none data from stationID 1 (probably
 not in use)
[*] 192.168.1.129:502 - Received: incorrect/none data from stationID 2 (probably
 not in use)
[*] 192.168.1.129:502 - Received: incorrect/none data from stationID 3 (probably
 not in use)
[*] 192.168.1.129:502 - Received: incorrect/none data from stationID 4 (probably
 not in use)
[*] 192.168.1.129:502 - Received: incorrect/none data from stationID 5 (probably
 not in use)
[*] 192.168.1.129:502 - Received: incorrect/none data from stationID 6 (probably
 not in use)
[*] 192.168.1.129:502 - Received: incorrect/none data from stationID 7 (probably
 not in use)
[+] 192.168.1.129:502 - Received: correct MODBUS/TCP from stationID  8
[*] 192.168.1.129:502 - Received: incorrect/none data from stationID 9 (probably
 not in use)
```

Figure 11.4: Screenshot showing Metasploit output





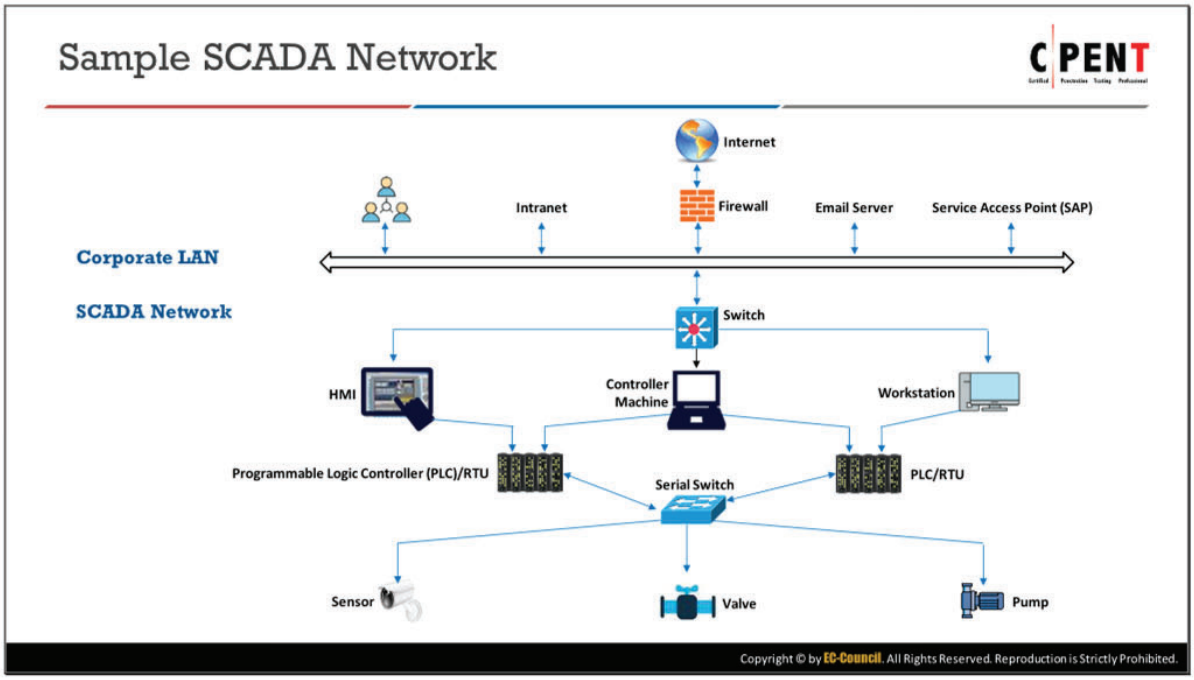
ICS and SCADA Pen Testing

The process of penetrating testing with ICS and SCADA is not the same as that of a normal IT pen test. With ICS/SCADA, penetration testers must determine the attack surface largely without sending data into the target, which results in a different type of process to follow for testing ICS and SCADA systems and, consequently, OT networks.

With this type of testing, penetration testers must ensure that they have accurate asset inventory and identification, this can be obtained from the client in the initial data call or from Open Source Intelligence gathering. SCADA penetration testing should be limited to test-bed or development systems and executed in a passive manner to avoid disrupting operations.

Step	Goal	Description
Asset and System Identification	▪ Determine threat vector and prioritize	▪ What will be assessed, the method, and in what order
Vulnerability Identification	▪ Determine the threat level and vector; map the attack surface and risk	▪ Identify the vulnerabilities that the discovered attack surface represents
Risk Determination	▪ Determine the possible impact	▪ Determine the risk based on the level of the discovered vulnerability

Table 11.2: Steps involved in ICS and SCADA pen testing



Sample SCADA Network

As shown in the figure, a corporate and a SCADA network are separated by a firewall. For external testing, the process is the same as for any other network, i.e., the filtering is the OT and IT networks must be determined. Like in most filter devices, the penetration tester must determine the rulesets and look for a specific vulnerability within them or accept the fact that the filters determine the attack surface and the risk from that attack surface; this is what should be ascertained during testing.

- **HMI/controller machine:** Usually, a Windows workstation is used as the master to manage and control PLCs on the network through client software. If compromised, an attacker can gain access to everything in the SCADA network. With the lifecycle of industrial systems being about 20 years, these Windows systems are often old. Therefore, it is critical for the client to understand the principle of least service and least privileges. The client mitigates and reduces their risk through penetration testing.
- **PLC:** A PLC is a physical system connected with a power supply and the capability to communicate over Ethernet networks. It could have an LCD panel showing the controller status, operator messages, etc. In recent times, PLCs have been shown to be accessible via web browsers, Telnet, and Secure Shell (SSH), exposing it to different kinds of application and network layer attacks. If compromised, an attacker can manipulate the I/O of the client's devices and cause serious damage to the organization. The PLC and the communication between it and the controller have been aspects that attackers could exploit; the famous Stuxnet attack was on a PLC.
- **End devices (sensor, valve, or pump):** End devices are installed at a remote site. They report to PLCs over communication links such as radio, serial connections, Ethernet, or

direct modems. If compromised, an attacker can compromise the integrity of the environment.

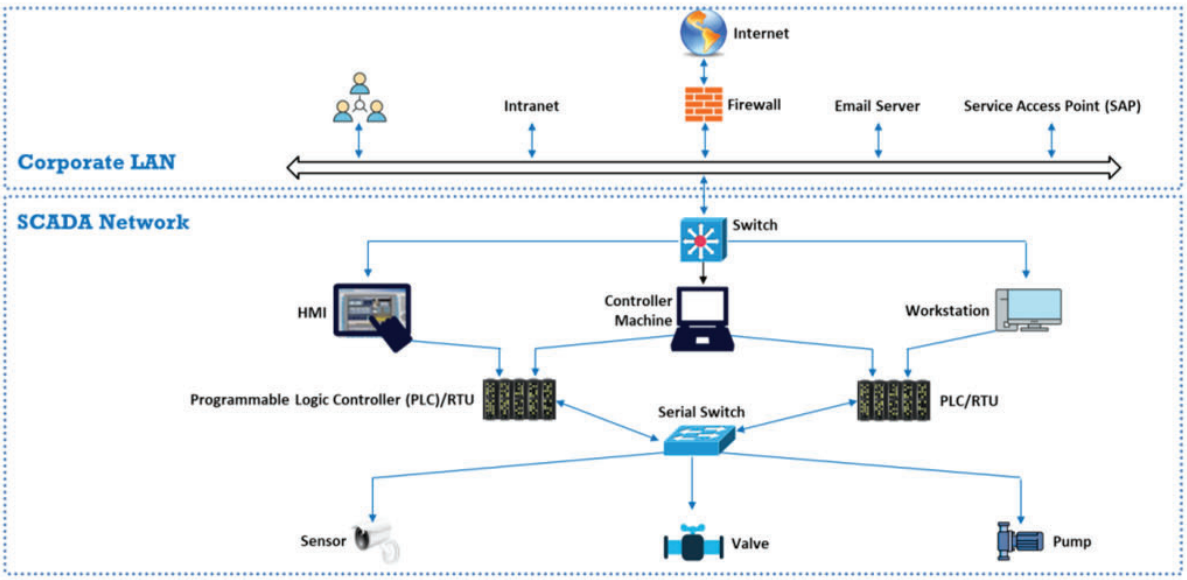
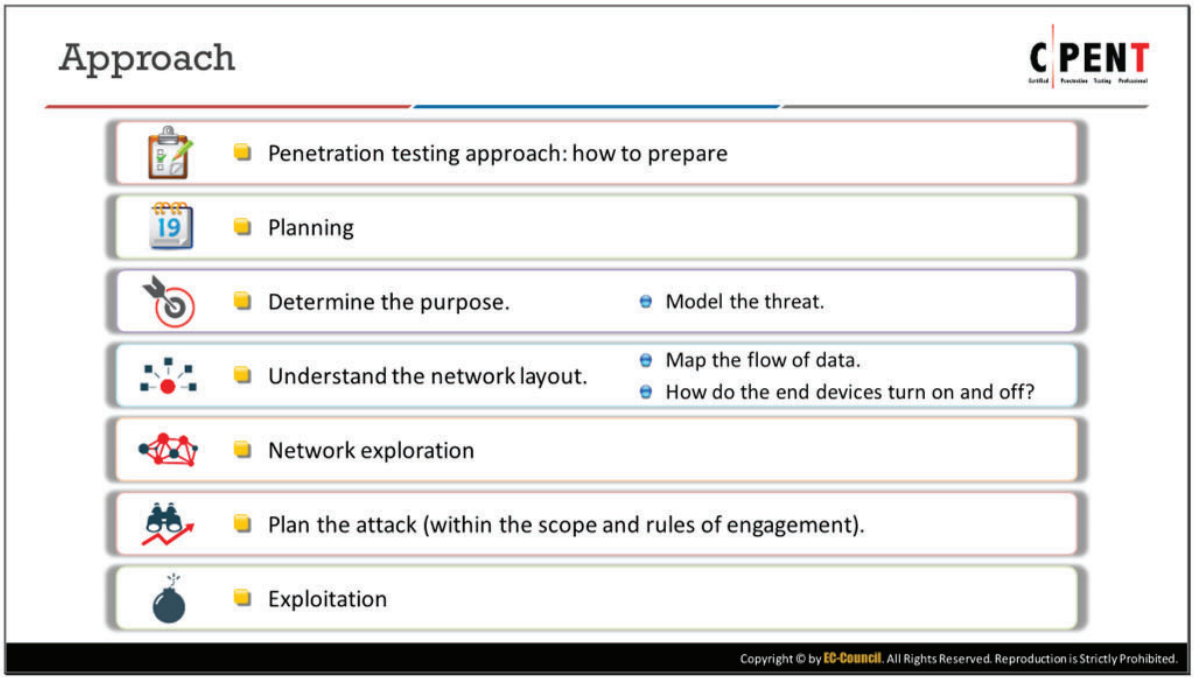


Figure 11.5: Illustration of SCADA network



Approach

There is only a slim chance that the client's site will have a test lab or mockup of their network. Nevertheless, penetration testers can attempt to emulate or build as much of a network to test on as possible to limit the impact on the site. However, some testing will need to be performed on the production side of the network, but as a best practice, the tester should avoid scanning anything on the production side and isolate any areas that control high-risk equipment. The tester must ensure that all stakeholders receive communications during each phase of testing.

- **Purpose**

  In IT testing, it is straightforward to determine the purpose of the target site. However, in OT, ICS, and SCADA testing, this is not so simple, because there are many different types of these networks running a variety of protocols. As the first prerequisite of any assessment, the pen tester must understand the purpose of the SCADA system: what mission-critical systems are in place, what functionality it provides, who the end users are, and how it benefits the organization. At this stage, it is a good idea to study system documentation, conduct research on implemented products and vendors, find known product vulnerabilities, and note factory-default credentials. A component of this stage is to model the threat.

- **Understand the network layout**

  The primary purpose of studying the network architecture is to logically understand how each component of the SCADA environment relates to another (this is highly complex). The penetration tester should understand what components are involved and how are they segregated, connected, or exposed to the wider network. This phase also involves the identification of various subnets present within the network. Furthermore, it is

important to determine whether the corporate network is separated from the SCADA network. The tester must mentally start to think about the attack surface; the presence of a web server; the presence of a database; and how these components connect, authenticate, and communicate. Moreover, the tester must consider the presence of applications visible at the front end, whether they can be fuzzed, etc.

- **Network exploration**

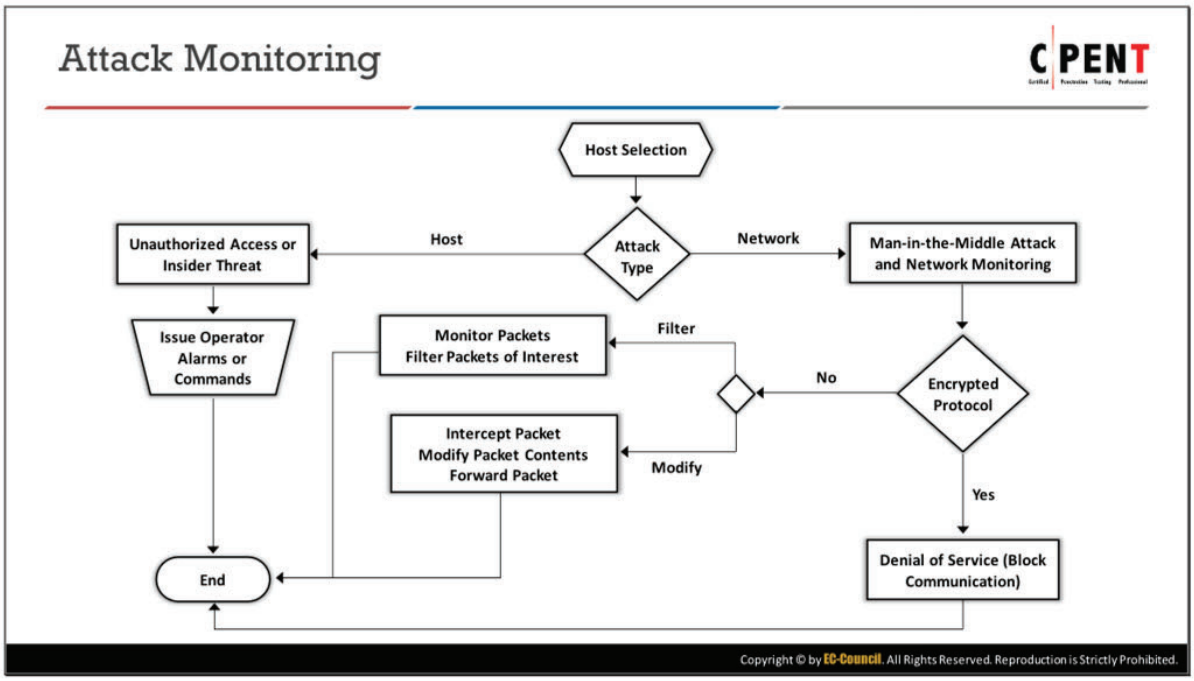
  With the information obtained from the preceding phases, the penetration tester can begin exploring the network. Network exploration must be performed with care and customer consent. If the systems and services are scanned, only one port and one target should be scanned at a time. However, in most cases, the tester should attempt to obtain this information passively. Wireshark and other protocol analyzers are powerful tools for this purpose. SCADA systems can be very fragile; therefore, the tester should exercise great caution and perform only well-bounded testing. They should perform exploration while keeping stakeholders in informed.

- **Plan attacks**

  The previous phase should provide the penetration tester sufficient information on what they need to test and what attacks are applicable. If attacks are allowed within the scope, then the tester must prepare test cases and ideally practice them before running them against the site assets. Furthermore, they should only use code that they wrote and customized or thoroughly tested in the lab. This is advisable for all forms of testing and not just for SCADA.

- **Exploitation**

  They client may ask the pen tester to execute each exploit individually. This will help in detecting the root cause in case any device unexpectedly experiences failure conditions. If failure occurs, the tester should halt testing and inform the client. Furthermore, they should attempt exploiting each component within the SCADA network, i.e., the network infrastructure, web interfaces, host operating systems (OSes), PLCs, HMI, and workstations, as in a conventional network pen test.



Attack Monitoring

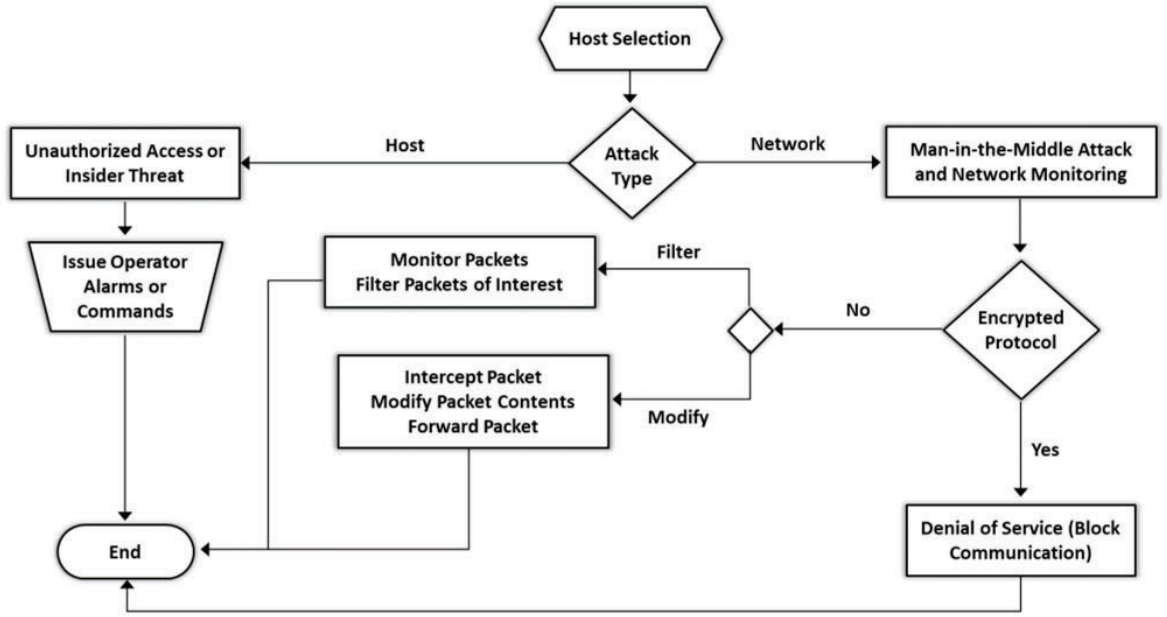


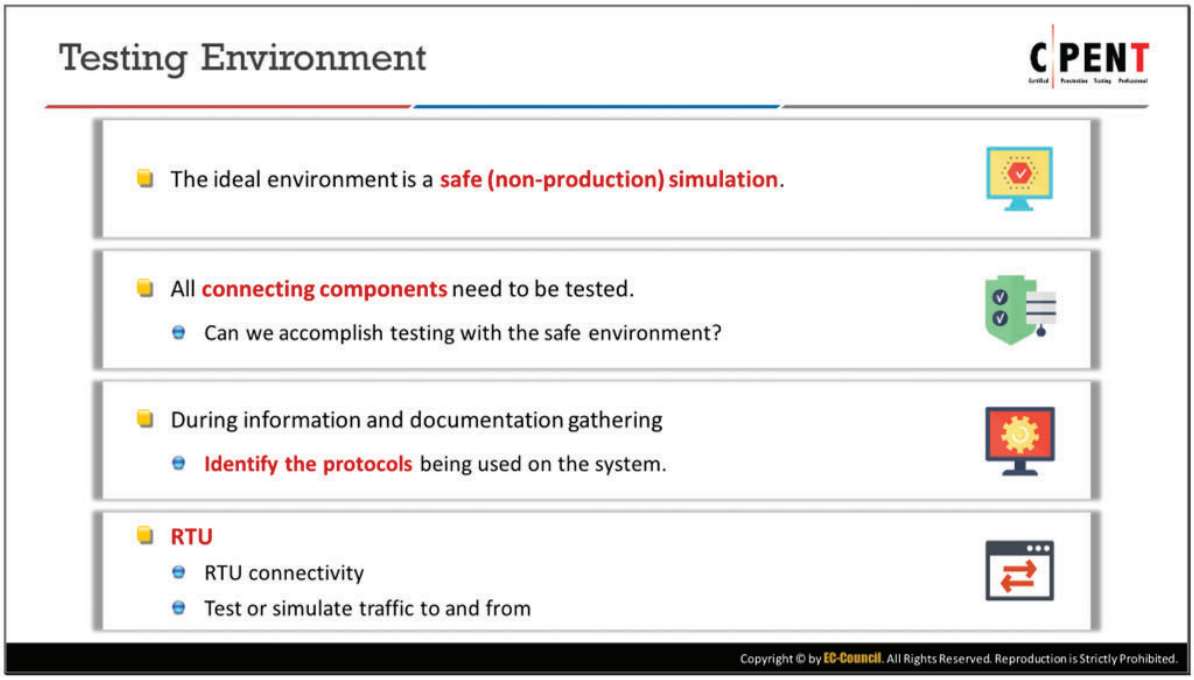
Figure 11.6: Illustration of attack monitoring

This diagram is based on a threat target of evaluation in which a man-in-the-middle (MITM) attack may be simulated to change the commands to and from the HMI merely to observe what occurs. The pen tester can modify and read the messages transmitted between the machines, similar to how the Modbus protocol can be manipulated.

As shown in the diagram, this level involves much of the same attack surface and vectors as in IT networks. The difference here is that the tester has access to data being sent; in other words, they can control end devices, which could result in serious damage or even loss of life.

This flow chart is a representation of the things that will more than likely be available when testing an OT network. Note that the flow chart may or may not be flat. Therefore, there could be a device between the pen tester and the network, which may limit or complicate what they can when testing. If the tester discovers from an external position that they can only map the attack surface through the device, which is usually a router or a firewall, then that is good for security. The tester's job is to map the attack surface and then determine the risk from that surface; they can use some tricks or bypass firewalls to penetrate the filtering device between them and the target. The good news is that most of the testing will be on the OT flat side of the network because the majority of the networks have all of its interfaces on that side. While this is often the case, to the tester must be prepared to identify any mistake and determine the existence of an interface into the OT from the IT, which is something that does occur.

The flow chart has an encryption encounter as a result of having to perform a denial-of-service (DoS) attack. However, penetration testers rarely resort to DoS attacks; instead, they examine the encryption protocol and algorithm being used and determine whether it is a strong or weak one, which would allow them to evaluate how to proceed.



Testing Environment

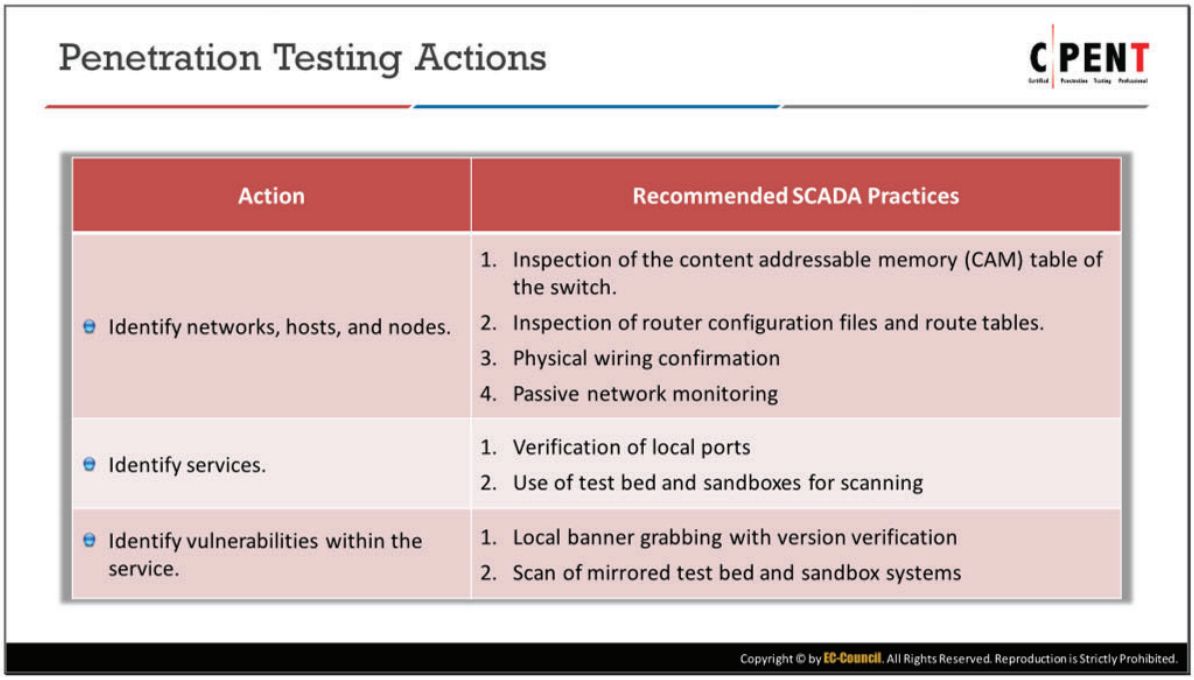
The ideal environment to assess SCADA is a safe (nonproduction) configuration. However, it is also desirable to have all connecting components and functionality available to confidently assess full system interoperability.

An RTU is used in SCADA systems at a remote location to collect code and transmit data back to the control station, as well as to receive and implement commands from the control center. Ideally, these components are configured with actual signals for the assessment, although emulators and simulators may be necessary.

The ability to establish a typical SCADA installation allows the assessment team to test and make recommendations on the configuration and deployment of a production system more accurately. Understanding the system configuration is key when deciding where the largest risks in the system may reside. Mirroring the connections to external systems is vital when replicating this configuration.

The assessment team must know where and how a SCADA system element typically connects to, for example, the Internet. Any of these elements being accessed from outside the SCADA network, such as the historian database, should be tested based on its position in the network.

Often, SCADA systems use OSes and hardware that are not in wide usage. Therefore, it is important that the assessment team have access to some of these less common platforms so that they can build their own testing environment for use during an assessment.

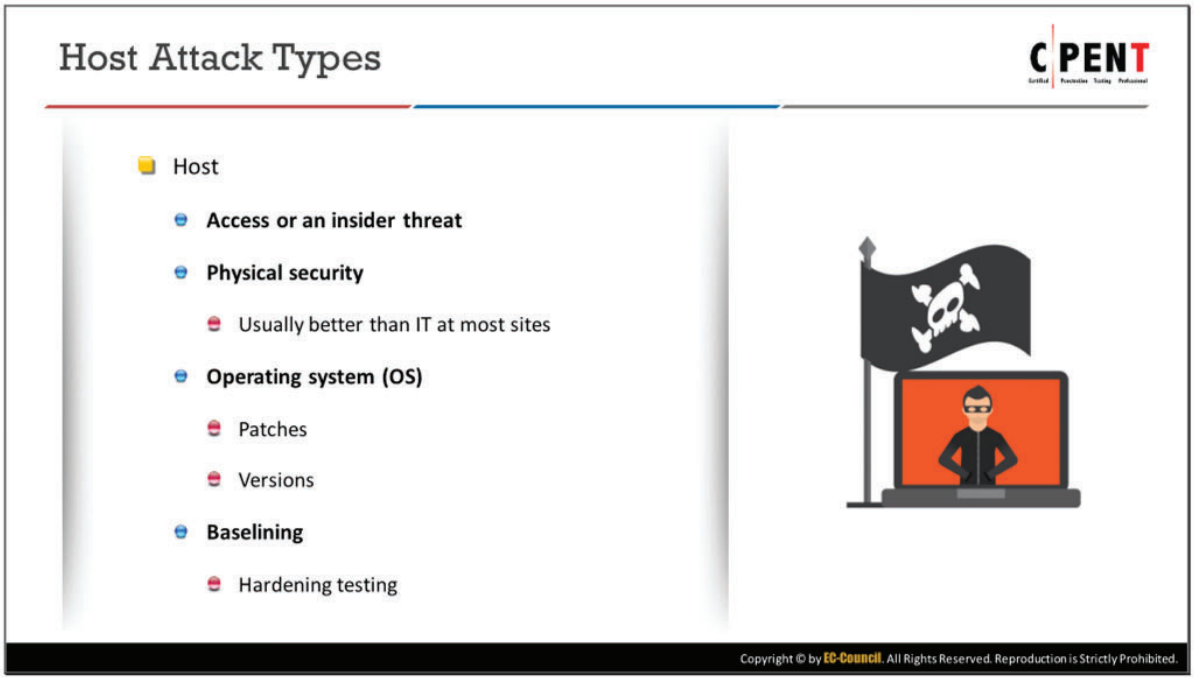


Penetration Testing Actions

A solid comprehension of the SCADA architecture that a pen test is intended for will help avoid pitfalls that could result in a negative effect on the operational process (i.e., hanging or crashing all or part of the process, costing the business lost revenue at the least or loss of life at the worst). The pen testing of IT systems or networks is often performed in real time. SCADA systems on an OT network should be examined as passively as possible. Preferably, there should be no active scanning or probing of the systems on the OT network to avoid compromising operations. In an ideal situation, a test bed, development, or sandbox system identical (e.g., mirrored) to the operational system would be in place not only for cybersecurity purposes but also for software testing and other implementation tests intended, eventually, for the OT system.

Action	Recommended SCADA Practices
▪ Identify networks, hosts, and nodes.	1. Inspection of the content addressable memory (CAM) table of the switch. 2. Inspection of router configuration files and route tables. 3. Physical wiring confirmation 4. Passive network monitoring
▪ Identify services.	1. Verification of local ports 2. Use of test bed and sandboxes for scanning
▪ Identify vulnerabilities within the service.	1. Local banner grabbing with version verification 2. Scan of mirrored test bed and sandbox systems

Table 11.3: Penetration testing actions and recommendations

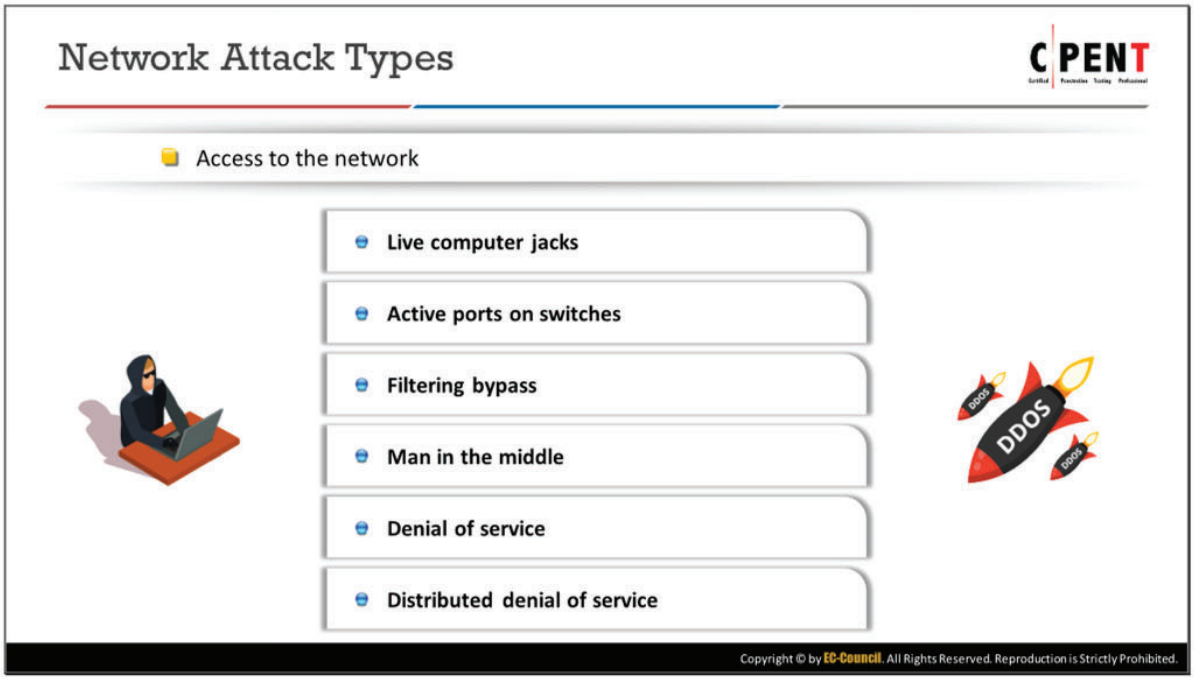


Host Attack Types

There is nothing new in the list as shown in the slide; many of the components of OT pen testing are the same as in the IT side.

As with all tests, physical security is a critical component. As the saying goes, "if I touch a box, I can own it." The OS is like anything else in IT testing. The pen tester should determine what OS the client uses and at what patch level, and in alignment with this information, they should determine if the client has and uses a patch management system. Indeed, some readers might ask why they should worry about these aspects when on offense. The answer is that the pen tester must improve the security and defenses of the client; ultimately, as a security expert, it is the job of the pen tester to improve the security process of their clients in the best way possible.

Therefore, when possible, the pen tester should ask about baselining and the vulnerability management program. It is a good practice to test a scenario of a vulnerability in a component and let the client show how they would handle it.



Network Attack Types

Once on site, it is imperative for network testing that the pen tester either obtain access to the switch room to determine the existence of ports that are not connected but are still live as well as to determine the existence of "live" network jacks anywhere in the site. This is a common problem because many in the OT field either have limited security knowledge or more likely consider their network isolated.

Next, the tester should consider their testing location. If the location is internal, then the tester cannot evaluate the attack surface through the perimeter device. If possible, the tester should obtain a network diagram and review the ingress and egress points. A site representative should hopefully be able to assist by revealing some aspects about their network as well as the traffic into and out of the segments. Hopefully, the client would have segments that help isolate traffic. However, the tester should not expect to obtain too much information because network engineers that know enough about their networks are rare even in the IT sector.

If the location is external, then the tester should attempt using some techniques to bypass the filtering. They can test for fragmentation scanning to check whether it yields different results, but it should only be performed on one port at a time and on noncritical points. The tester can use source port scanning to find any weak rules. Since these networks are usually on a very long lifecycle, there are chances of finding weak filtering; thus, they should be tested.

For internal testing, the following levels of access must be tested for:

- Unauthenticated
- Authenticated as a regular user
- Authenticated as a privileged user

In most cases, the tester will have more than enough findings with the first two and rarely need the third item.

For MITM attacks, it is best to run Wireshark or another protocol analyzer and determine whether the network is well segmented with switches and proper virtual local area network (VLAN) configurations. Another trick is to spoof the media access control (MAC) address of the default gateway and determine the existence of any traffic going in and out of the OT network that is not protected with first encryption and authentication. Since some of these networks are quite dated, there is a chance of finding cleartext protocols such as the File Transfer Protocol (FTP) and Hypertext Transfer Protocol (HTTP), which should be assessed.

Another task is to determine the possibility of egressing out from the OT network with an encrypted protocol such as SSH or any virtual private network (VPN) protocol; in such a case, the tester can blind any monitoring the client has in place. If the tester discovers through MITM attacks that the client has done a good job and encrypted the traffic, they should attempt to spoof the encryption and get the user to either accept the certificate, if they are using HTTP secure (HTTPS), or store the key, if they are using SSH.

If possible, the pen tester should obtain configurations and review them for the best practices of ingress and egress filtering as well as the recommended guidelines to defeat distributed DoS attacks.

The tester should assess the environment and test for the following:

- Ingress
- Sanity checking
- Bogon filtering and RFC 2827 to defeat DoS attacks
- RFC 3704 and GEO IP blocking
- Egress
- Blocking of outbound connections initiated from servers—on discovering servers, spoof the network traffic and determine if it is allowed out or if the access to other things that were blocked before are now allowed.
- Blocking of any packet that does not have a source IP address of the site network
- NIST 800-82 best practices
- Blackhole routing—is the network traffic for the Domain Name System (DNS) and HTTP/HTTPS from one source or can anyone directly egress out?

If the tester cannot gain access to the configurations of the devices between the OT and IT, then they should test it from both sides with their team.

Ports of SCADA

Protocol	Ports
BACnet/IP	UDP/47808
DNP3	TCP/20000, UDP/20000
EtherCAT	UDP/34980
Ethernet/IP	TCP/44818, UDP/2222, UDP/44818
FL-net	UDP/55000 to 55003
Foundation Fieldbus HSE	TCP/1089 to1091, UDP/1089 to 1091
ICCP	TCP/102
Modbus TCP	TCP/502
OPC UA Binary	Vendor Application Specific
OPC UA Discovery Server	TCP/4840
OPC UA XML	TCP/80, TCP/443
PROFINET	TCP/34964 to 34964, UDP/34962 to 34964
ROC PLus	TCP/UDP 4000

Ports of SCADA

The concern with Internet-facing PLCs is that they can be targeted by adversaries to breach the perimeter and enter the network. Then, they can attempt to achieve persistence and start scanning devices over the network, move laterally across, and perform malicious actions, such as bringing down the plant or creating a disaster.

The following are some of the threats for Internet-facing PLCs:

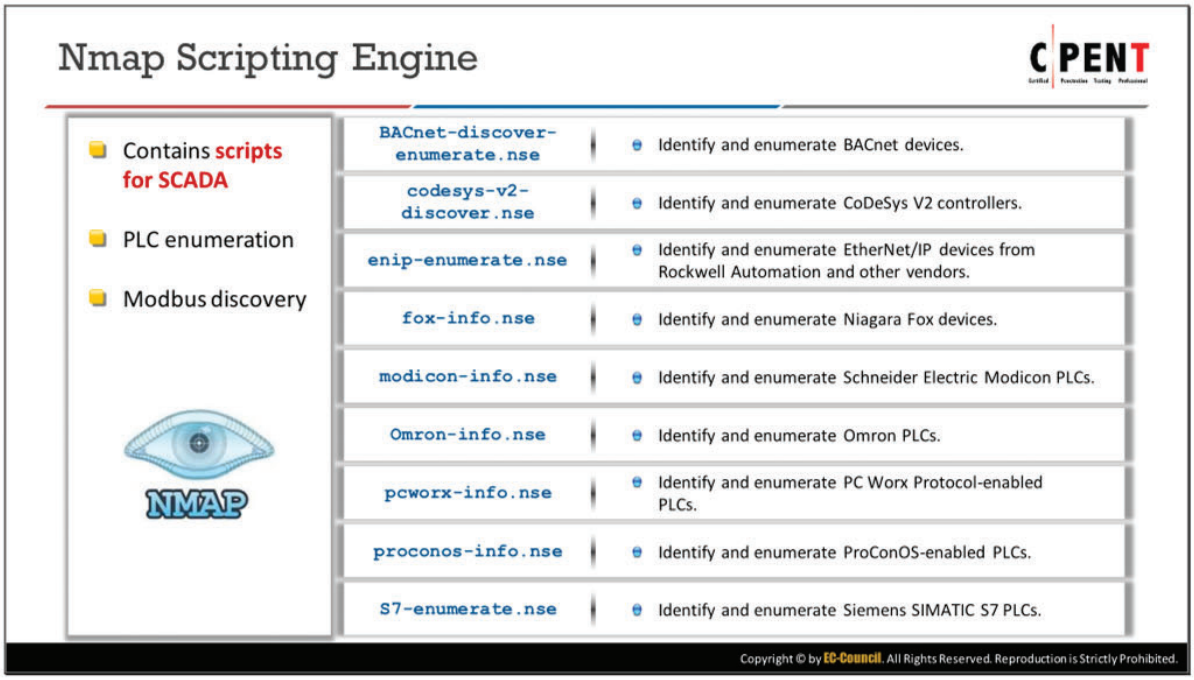
- Remote-access attacks
- PLC worms; payload sabotage attacks
- PLC rootkits

For PLCs that are directly accessible, even though only the PLC web server is accessible directly from the Internet, several remote-access attacks could be performed, including DoS attacks and integrity attacks such as downloading and overwriting of the PLC program with a sabotage PLC program. Doing so compromises the integrity of the PLC. For instance, port 102 is used by the Siemens proprietary S7 family PLCs, and port 502 is used for Modbus, which is one of the de facto industrial control system protocols for communication between a PLC and a plant component acting as the slave device. Each slave in Modbus has a unit ID, which gives an idea of how many slaves are being controlled by that PLC.

Nmap now has scripts for SCADA, and certain Nmap scripts such as "`s7-info.nse`" are written specifically for ICS environments to gather information about Siemens PLCs. Therefore, the PLC is one of the main components to test.

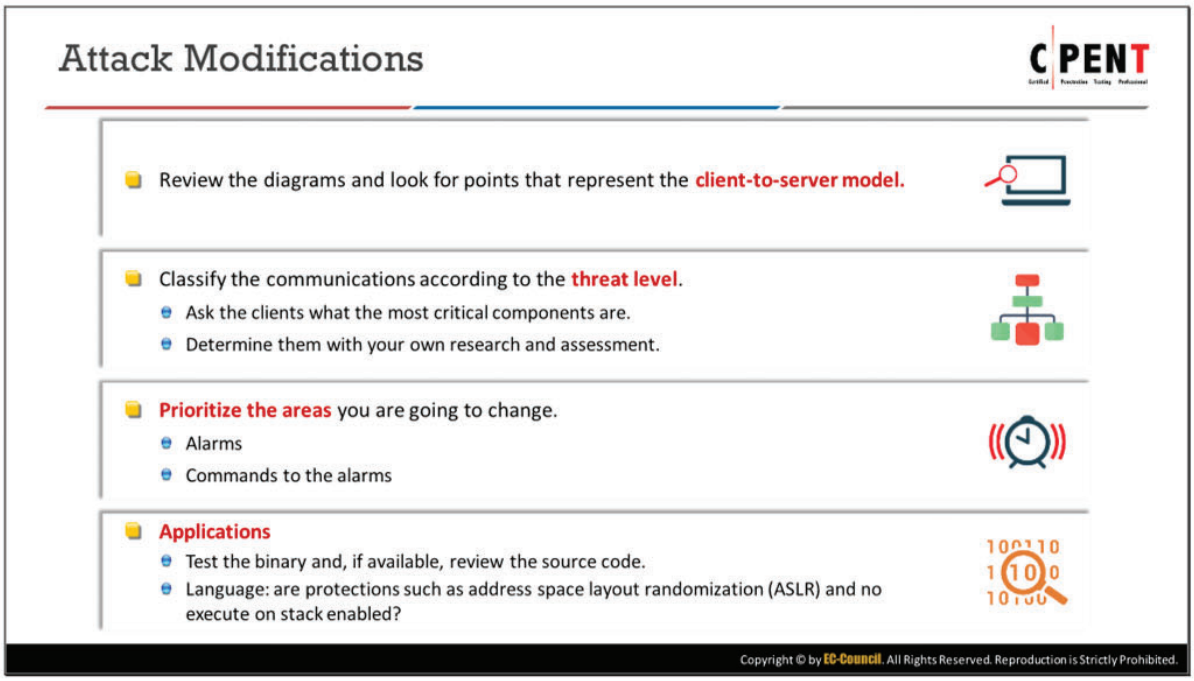
Protocol	Ports
BACnet/IP	UDP/47808
DNP3	TCP/20000, UDP/20000
EtherCAT	UDP/34980
Ethernet/IP	TCP/44818, UDP/2222, UDP/44818
FL-net	UDP/55000 to 55003
Foundation Fieldbus HSE	TCP/1089 to1091, UDP/1089 to 1091
ICCP	TCP/102
Modbus TCP	TCP/502
OPC UA Binary	Vendor Application Specific
OPC UA Discovery Server	TCP/4840
OPC UA XML	TCP/80, TCP/443
PROFINET	TCP/34964 to 34964, UDP/34962 to 34964
ROC PLus	TCP/UDP 4000

Table 11.4: Ports of SCADA



Nmap Scripting Engine

Quite a few scripts are available within the Nmap tool, and a tester should be aware of these.

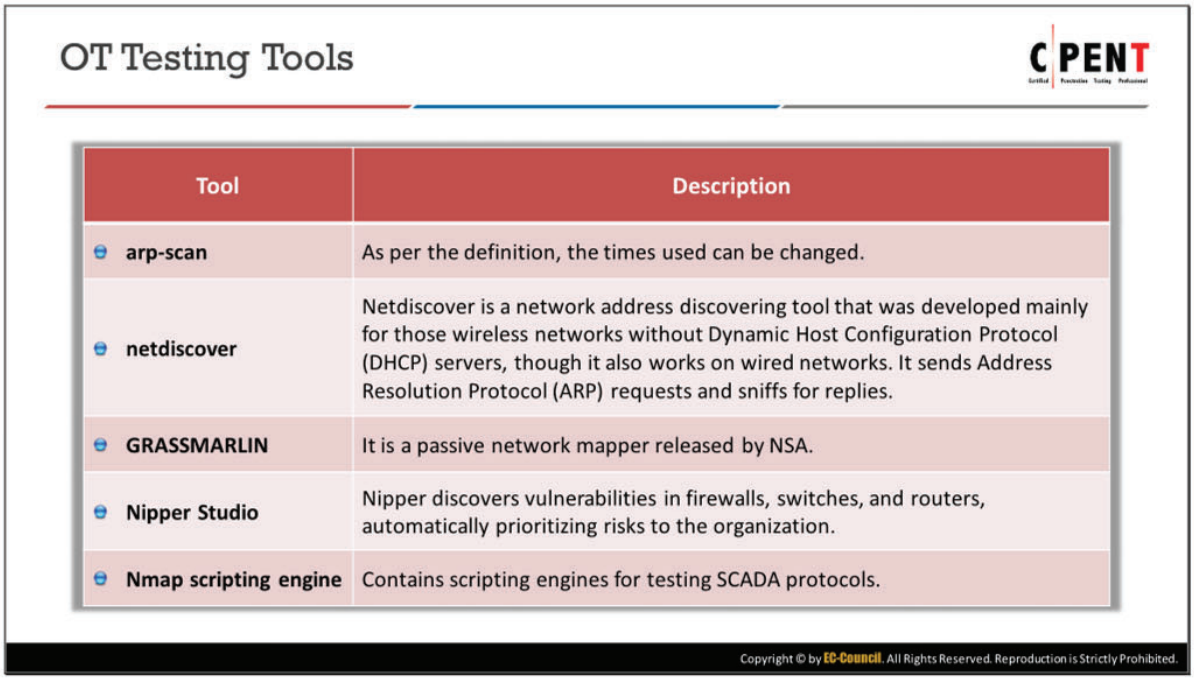


Attack Modifications

Attack modification is an aspect where the rules of engagement are critical because the pen tester must have a detailed understanding of what they can and cannot test. Before sending one packet on the OT side of the network, the tester must ensure that every precaution has been taken and that they have adequately defined the boundaries of testing, which will be discussed later.

The pen tester must create a detailed testing plan and review it with the most technical representative from the client. After reviewing the site, the plan should suggest methods of attacking the site and the associated threat level; it is common to select a medium-level threat. A critical aspect here is the assistance from the client. The pen tester must have a listing of the data requirements to conduct the test. If the client has a sample site or testbed, which is rare, then this aspect is not as critical; however, it is absolutely critical for testing a live site.

With the above in mind, the tester must establish a recovery plan and a procedure to follow in case a system gets corrupted while testing. A well-defined assessment plan provides detailed directions to the team, eliminating confusion and wasted time.



OT Testing Tools

Many tools can be used within OT networks, several of which are the same as those used in IT networks.

- **arping**

  arping (*https://github.com/ThomasHabets/arping* or *https://github.com/iputils/iputils*) can be used to identify live hosts and devices by sending a single Address Resolution Protocol (ARP) request to a single target. ARP traffic is a native occurrence on all Ethernet networks that communicate via IP. Any ICS device with an IP address receives ARP requests on a semiregular basis and can handle such requests using this technique in moderation. One IP address should be requested at a time, as shown next, and requests should be separated by reasonable time intervals. In other words, an ICS network should not be subjected to an "ARP storm." Indeed, this can be tedious and time-consuming method; however, a script can be written to help automate the process.

  The following is an example of using a "one-liner" Bash script to automate requesting a list of IP addresses with pauses for 5 s after each request. The option **`-i eth0`** shows the interface, and **`ipaddresses.txt`** contains a list of IP addresses with one per line. The number of seconds to pause can be adjusted by changing **`sleep 5`** to any number of seconds as desired.

  ```
  while read ip; do arping -i eth0 -c 1 $ip; sleep 5; done < ipaddresses.txt
  ```

- **arp-scan**

  The arp-scan tool can be used to "scan" the entire subnet that corresponds to the designated network interface. Note that this tool designates time in milliseconds, sending requests every 5000 ms (`-i 5000`), which can be changed as desired.

  The `arp-scan` command can also specifically designate a network range to scan using the Classless Inter-Domain Routing (CIDR) notation (`192.168.50.0/ 24`), and it does not necessarily have to be configured on the local network interface. Therefore, this tool is very useful for scanning general network ranges without actually having an IP address on the target network.

  See *http://www.blackmoreops.com/2015/12/31/use-arp-scan-to-find-hidden-devices-in-your-network/* for a tutorial on the use of this tool to discover hidden devices.

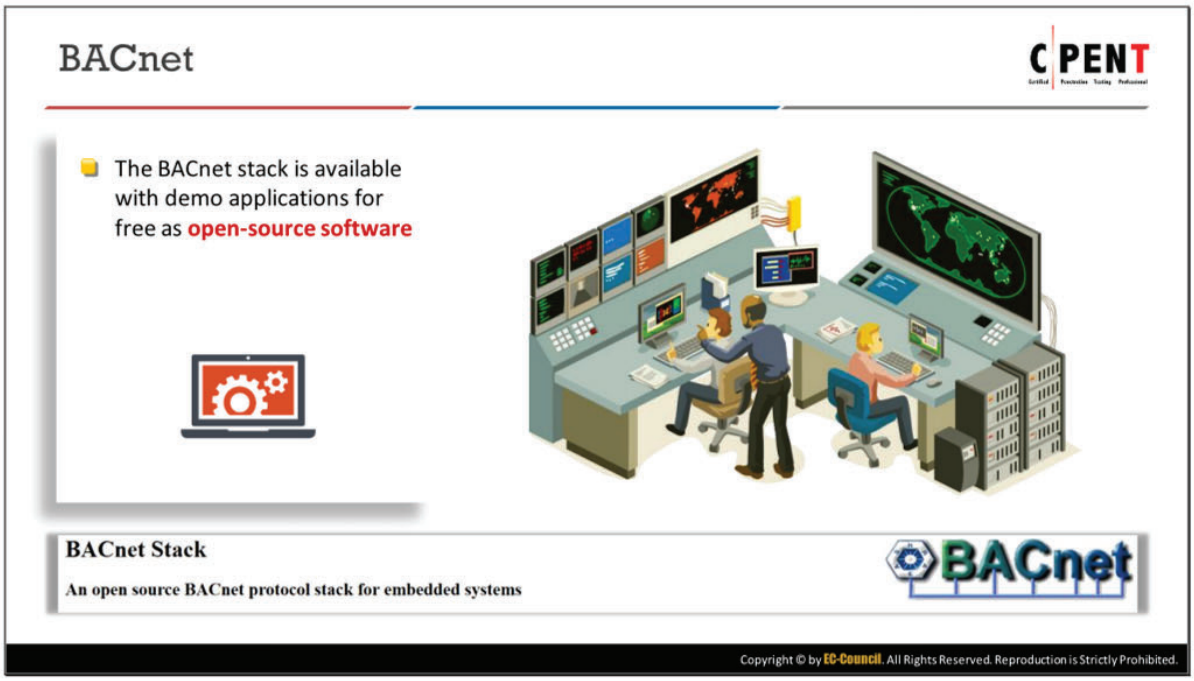
  One of the shortfalls of both arping and arp-scan is that they only reveal whether there is a live host or device using a particular IP address. To determine what type of host or device is using the address, the tester must look up its MAC address using online tools such as the Wireshark OUI lookup tool (*https://www.wireshark.org/tools/oui-lookup.html*). The tester can also write a script that can query the entire OUI database (*http://standards-oui.ieee.org/oui.txt*) and compare it to their results.

- **GRASSMARLIN**

  In early 2016, the U.S. National Security Agency released GRASSMARLIN, which is a free passive network mapper specifically intended for industrial networks. It has recently become open source (*https://github.com/iadgov/GRASSMARLIN*) under the GNU Lesser General Public License Version 3 licensing. GRASSMARLIN provides a "snapshot" of the ICS environment, including devices on the network, communications between these devices, and metadata extracted from these communications.

- **Nipper Studio**

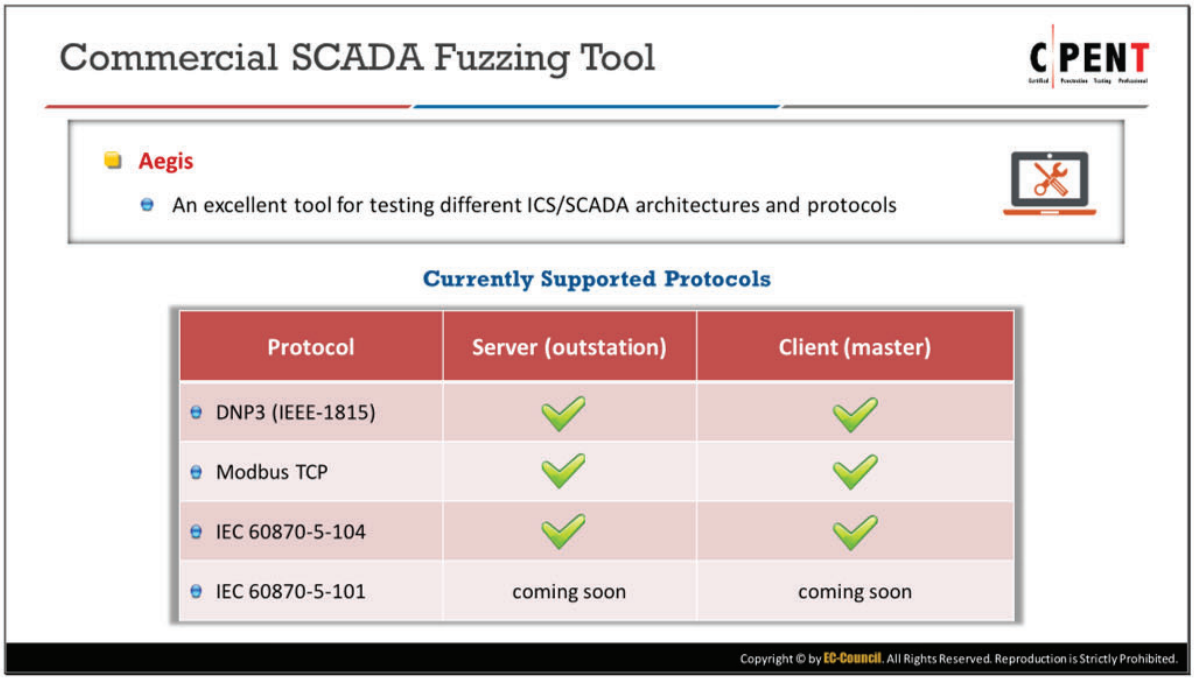
  Nipper Studio is an excellent tool for reviewing the configuration of devices. It enables testing configurations in accordance with best practices as well as finding any vulnerabilities associated with the configuration.



BACnet

BACnet is a standard data communication protocol for building automation and control networks. It is an open protocol, which means that anyone can contribute to the standard and anyone may use it. The only caveat is that the BACnet standard document itself is copyrighted by ASHRAE, which sells the document to help defray the costs of developing and maintaining the standard (similar to IEEE, ANSI, or ISO).

For software developers, the BACnet protocol defines a standard way to communicate with other BACnet-compliant devices over a number of wires known as data link/physical layers: Ethernet, EIA-485, EIA-232, ARCNET, and LonTalk. The BACnet standard also defines a standard way to communicate using UDP, IP, and HTTP (web services).

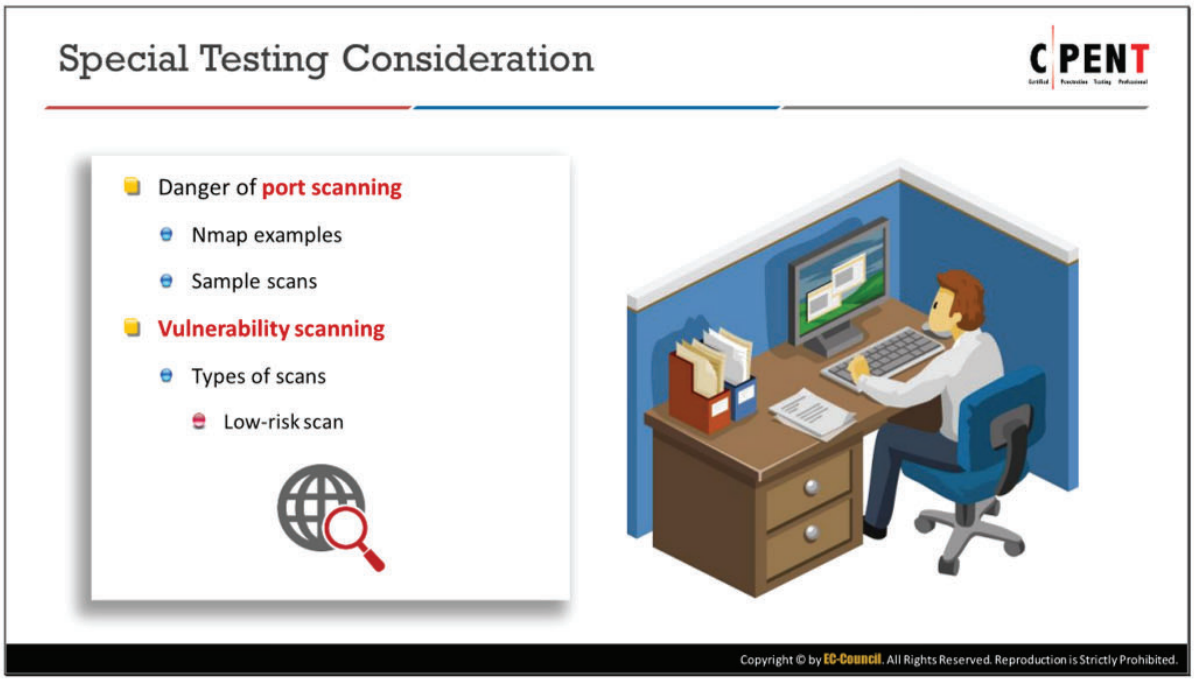


Commercial SCADA Fuzzing Tool

With a sufficient budget, the Aegis tool is excellent to work with and has a large database of alerts for different protocols.

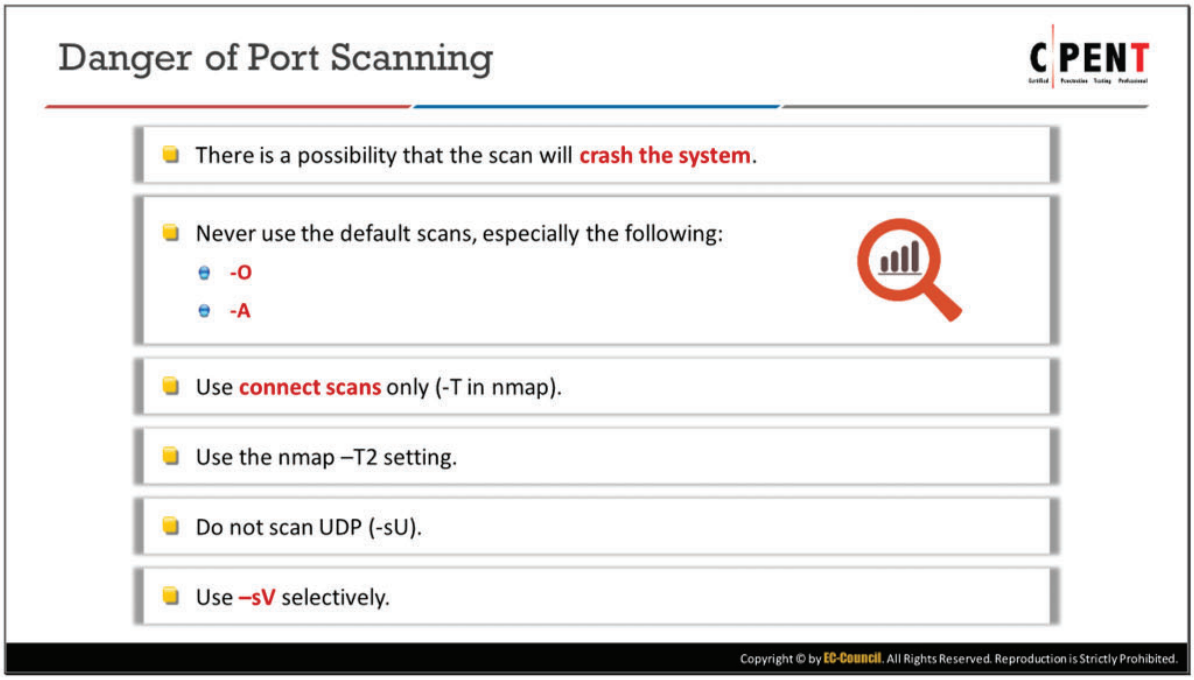
Protocol	Server (outstation)	Client (master)
DNP3 (IEEE-1815)	✓	✓
Modbus TCP	✓	✓
IEC 60870-5-104	✓	✓
IEC 60870-5-101	coming soon	coming soon

Table 11.5: Currently supported SCADA protocols



Special Testing Consideration

As mentioned previously, there are special considerations when scanning embedded systems. The penetration tester cannot use any of the default scan options, and all of the scanning they perform must be well bounded and tested in a lab environment thoroughly before implementation in a production environment. With vulnerability scanners, this is even more of a concern because the default profiles and policies are too intrusive on these networks. Therefore, it is best to not use any vulnerability scanner; it is recommended to create a custom scan instead. However, the tester should be guided by their testing and exploration research. Further, they should discuss with the client if possible; if not, then it is best to always error on the side of caution.



Danger of Port Scanning

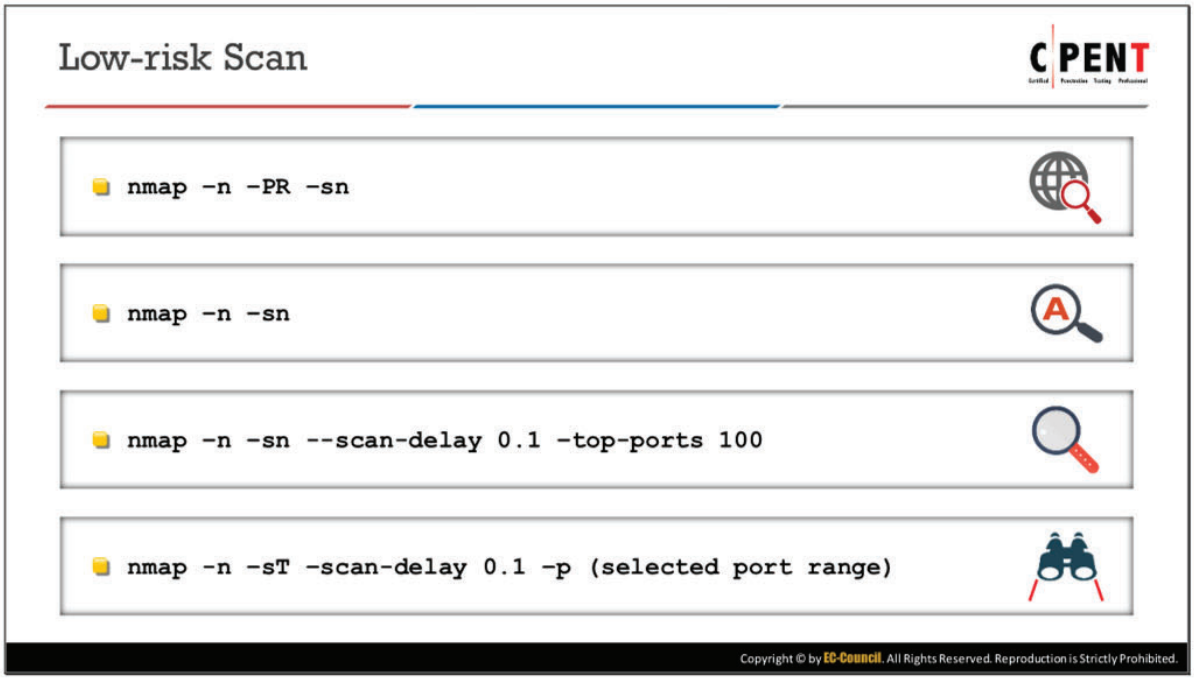
As discussed previously, at the other end of these client–server relationships is often a device that controls equipment. If the penetration tester scans this device, it can quickly become very dangerous; therefore, it is best to avoid scanning it at all costs and use slow and methodical scans of one or two ports at a time. Although such a scan will take a long time, it avoids crashing network components.

Recall that, when on the same subnet, Nmap does not use the Internet Control Message Protocol (ICMP) and instead uses ARP; therefore, there should be no problem with this since it uses the MAC address to identify the devices. Connect scans should be used because these systems are not designed for the **`-sS`** or SYN/stealth/half-open scan. When a destination receives a packet with the SYN flag set, the socket enters the SYN-RECEIVE state, also known as the half-open state. Subsequently, when the corresponding ACK flag that would send the socket into the ESTABLISHED state is not received, the connection maintains that open queue in memory. As a result, the connection consumes resources; the more resources it consumes, the more likely it is to crash the machine. Again, this is something to avoid. Therefore, a connect scan has a much lower chance of crashing the machines and/or devices.

The timing should be set with thorough consideration as well; penetration testers once scanned a telephone provider network at their request, and their bill payment machines all crashed from a simple Nmap default scan with nothing set other than **`-sS`**. Consequently, the telephone provider's machines all over the country stopped working, their customers could not make payments or perform other actions, and their help desk was completely overwhelmed with calls. To make matters worse, penetration testers had to write a report and submit it to the application developer to show what penetration testers did and the results. To be fair, most of the scans were against an old OS, but when the client gave penetration testers a sample of the new

machine with the latest OS, it failed as well. As a result, they had to do a complete code rewrite. Again, this is the job of testers, but the client did not define that the instructions they gave was for a scan of all the "live" machines in the country. In hindsight, it would have been best to avoid this situation with a test scan of one machine, but this was a very large project with many team members. Therefore, penetration testers must always ensure that the team is well briefed. Another option is to set **`--scan-delay`** to, say, 0.1 or to use **`–max-parallelism 1`** to scan one port at a time per host.

Scanning UDP ports with null payloads can affect ICS software on Windows and Linux as well. Service fingerprinting is usually safe, but it should first be tested, and in lieu of **`–sV`**, **`--script=banner`** can be used.



Low-risk Scan

Always run Nmap with the **sudo** command when working on *nix, e.g., **sudo -sT**. As with any scan, Nmap should be examined at the packet level. Further, although verbose should be enabled, it should not be relied upon.

nmap -n -PR -sn

This forces Nmap to only use ARP requests to find live hosts. Therefore, if the host answers, a live machine has been identified. ARP occurs so frequently in networks that it is considered normal and should not cause any problems. However, on large subnets, this could generate a large amount of noise and a possible ARP storm; thus, it is advisable to exercise caution. Moreover, this scan must be on an internal network since it uses ARP. Additionally, no DNS lookup or port scan should be performed; only the target should be scanned.

ARP Storm

The ARP traffic on a network can be around 10% of the network traffic during normal times and up to 85% of the network traffic during abnormal network conditions caused by an ARP broadcast storm. Basically, ARP requests are broadcast multiple times or to multiple systems (occasionally more than a thousand requests within a second) during an ARP broadcast storm, thereby consuming a lot of network bandwidth and making normal network communications difficult.

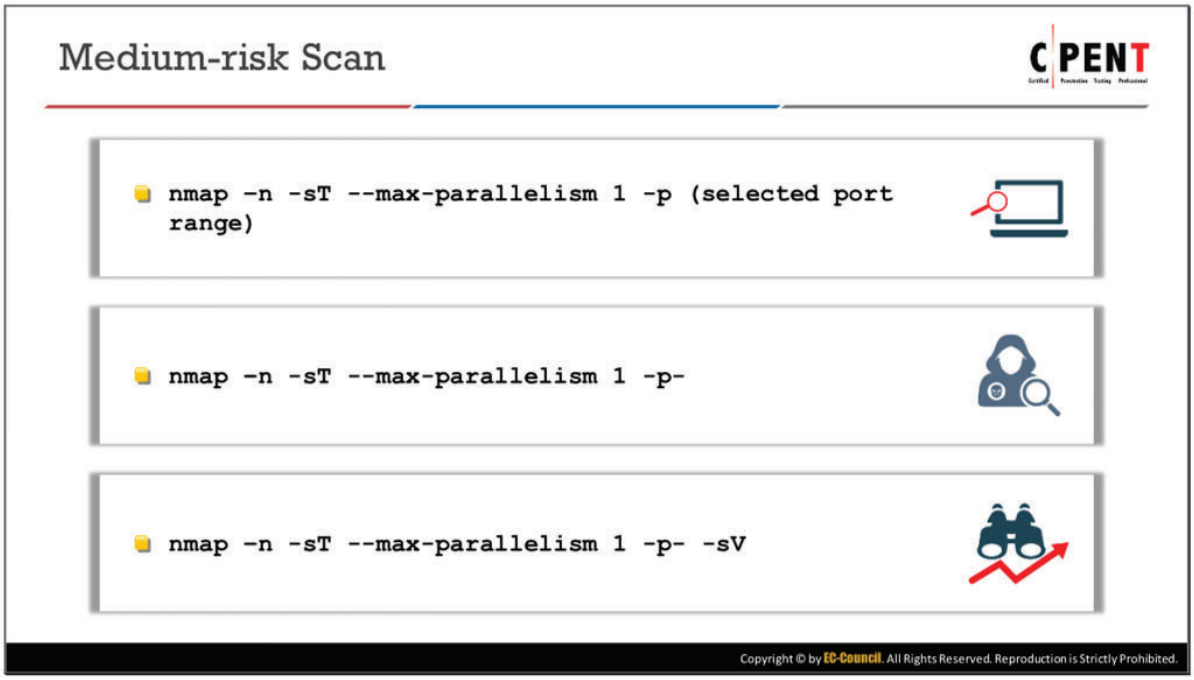
A large unsegmented network is especially susceptible to excessive ARP broadcasts/broadcast storms. Hence, it is always a good practice to sub-divide a large network into various segments. This limits the broadcast domain to a limited number of systems.

nmap -n -sn --scan-delay 0.1 -top-ports 100

Since there is no port scan (**-sn**), this command only sends the ICMP and TCP 80/443 ping requests. This will scan the ports in a serial manner with delays of 1 s in between.

`Nmap -n -sT -scan-delay 0.1 -p ???` …

Unlike the previous one, the above scan will cover all of the 1000 default ports serially and, again, with delay of 1 s in between. However, the concern here is that the scan might overwhelm a low-powered or fragile device. Therefore, this scan should be used with discretion. The **`-p`** option can be used to control how many ports the scanner scans, whereas in the previous example, the scan was for the list of the top 100 ports.



Medium-risk Scan

`nmap -n -sT --max-parallelism 1 -p (selected port range)`

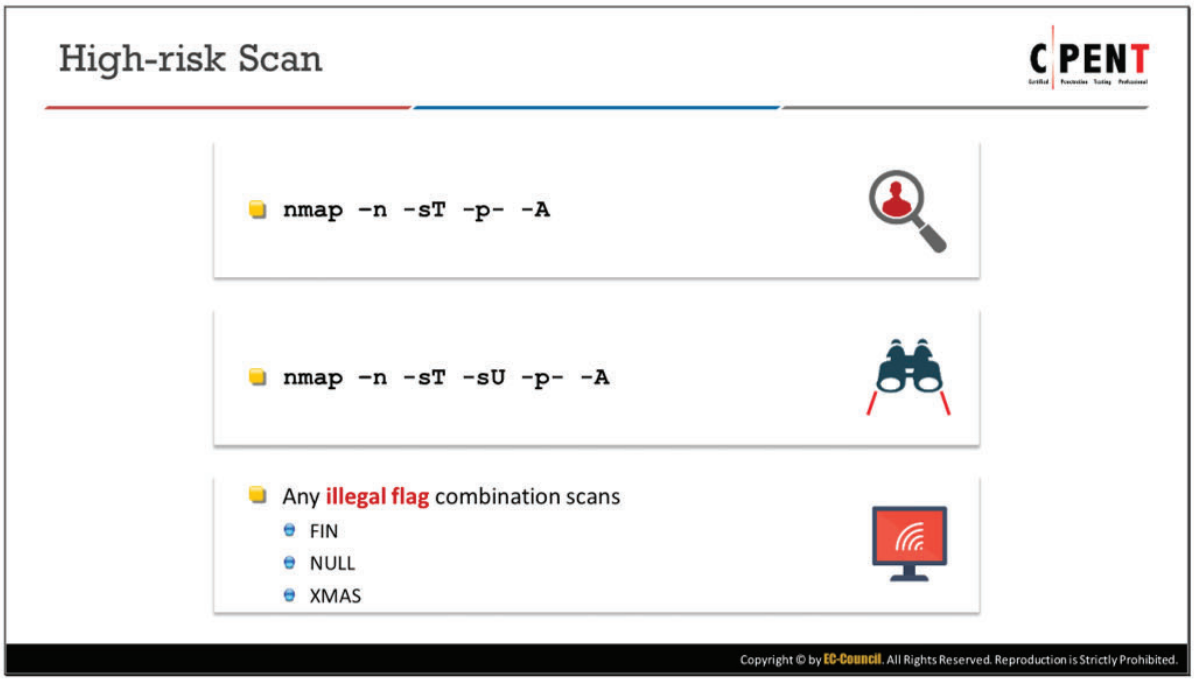
The above scan is considered a medium-risk port scan since it scans only one port at a time and the range can be controlled. This is the power of using custom scans. Based on the initial planning and research, the penetration tester can identify what to scan or not to scan; accordingly, they can run this scan while avoiding critical sectors and anything other than the low-risk scans discussed previously.

`-min-parallelism <numprobes>; --max-parallelism <numprobes> (Adjust probe parallelization)`

The above options control the total number of probes that may be outstanding for a host group. They are used for port scanning and host discovery. By default, Nmap calculates an ever-changing ideal parallelism based on network performance. If packets are being dropped, Nmap slows down and allows fewer outstanding probes. The ideal probe number slowly increases as the network proves itself worthy. These options place minimum or maximum bounds on that variable. By default, the ideal parallelism can drop to one if the network is unreliable or increase to several hundred under perfect conditions.

The most common usage is to set **`--min-parallelism`** to a number higher than one to speed up scans of poorly performing hosts or networks. This is a risky option to modify, as setting it too high may affect accuracy. Setting this also reduces Nmap's ability to control parallelism dynamically based on network conditions. A value of 10 might be reasonable, but this value should only be adjusted as a last resort. The **`--max-parallelism`** option is sometimes set to one to prevent Nmap from sending more than one probe at a time to hosts, which is how it is used when scanning these networks.

`nmap -n -sT --max-parallelism 1 -p-`



High-risk Scan

```
nmap -n -sT -p- -A
```

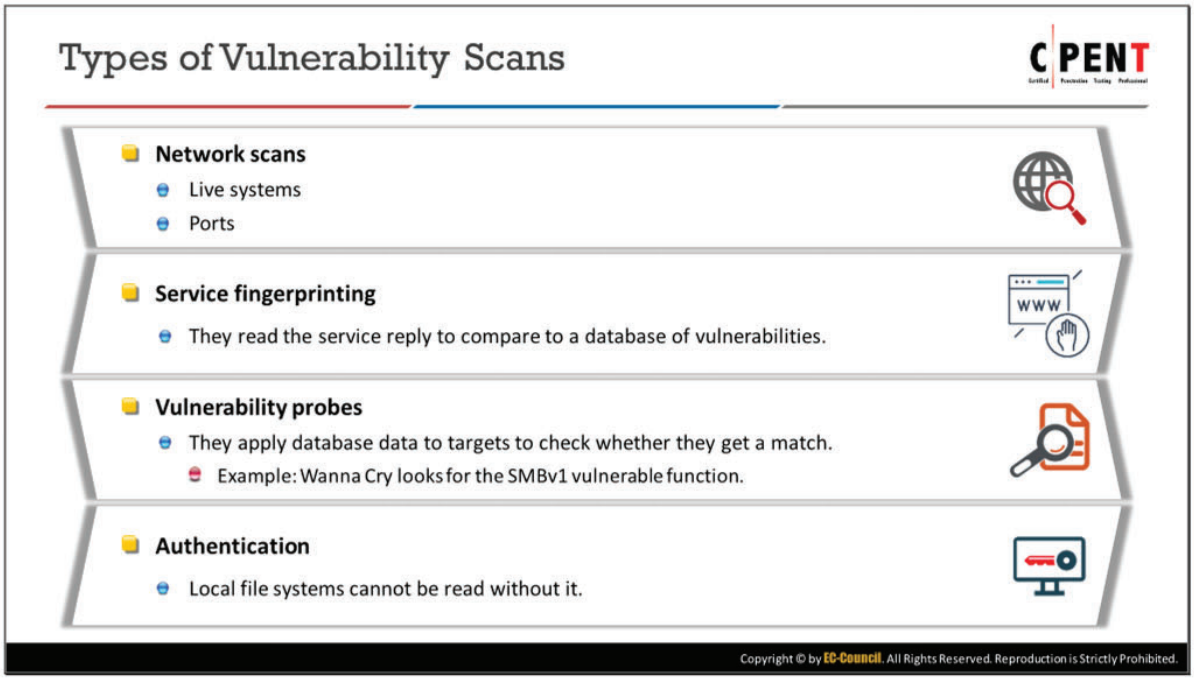
The above scan should never be used as it will more than likely crash old systems. It scans all possible ports and is of high risk. Additionally, it fingerprints everything and runs the Nmap scripting engine.

```
nmap -n -sT -sU -p- -A
```

Again, the above scan should not be used. Instead of just the TCP ports, the UDP ones have been added here.

Illegal flag combinations

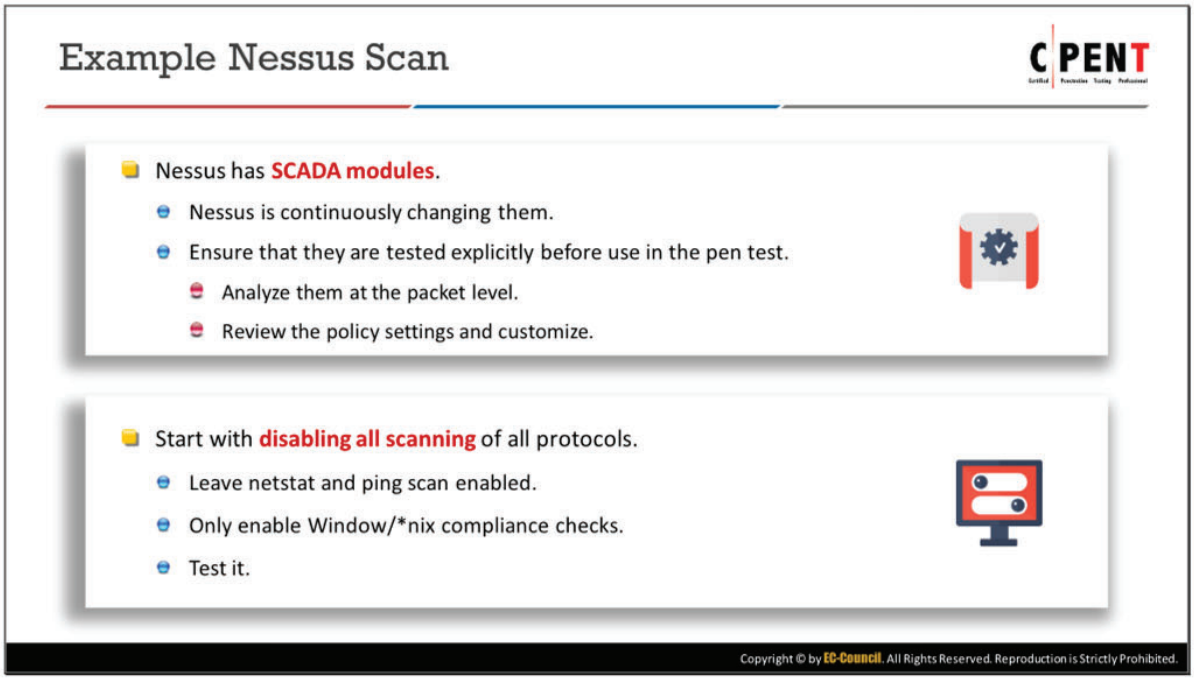
While illegal flag combinations are popular to discuss, they should not be used against any OT network assets, because they violate the RFC. Furthermore, OT networks can have old legacy systems that are not equipped with the defenses available today. Even simple attacks with illegal flag combinations such as the ping of death should never be attempted on OT networks and/or devices.



Types of Vulnerability Scans

Most vulnerability scanners rely on a database of vulnerabilities. As such, they only find the types of vulnerabilities that are known. Additionally, these scanners do not go through filters very well. They work best when used for internal testing, with the exception of web scanners, which scan the attack surface represented by the server and any applications that are running on that server; as a result, web scanners are designed for testing from perimeter locations. Another aspect of vulnerability scanners is their capability to read the local file system, for which they need credentials. For *nix systems, the scanners mostly use SSH, while for Windows-based systems, they use the Server Message Block (SMB) protocol to gain access. Even web scanners perform better once they have authentication capability; without it, only the first layers of the web logic can be tested. Recall that, for authentication, three levels of permission are required:

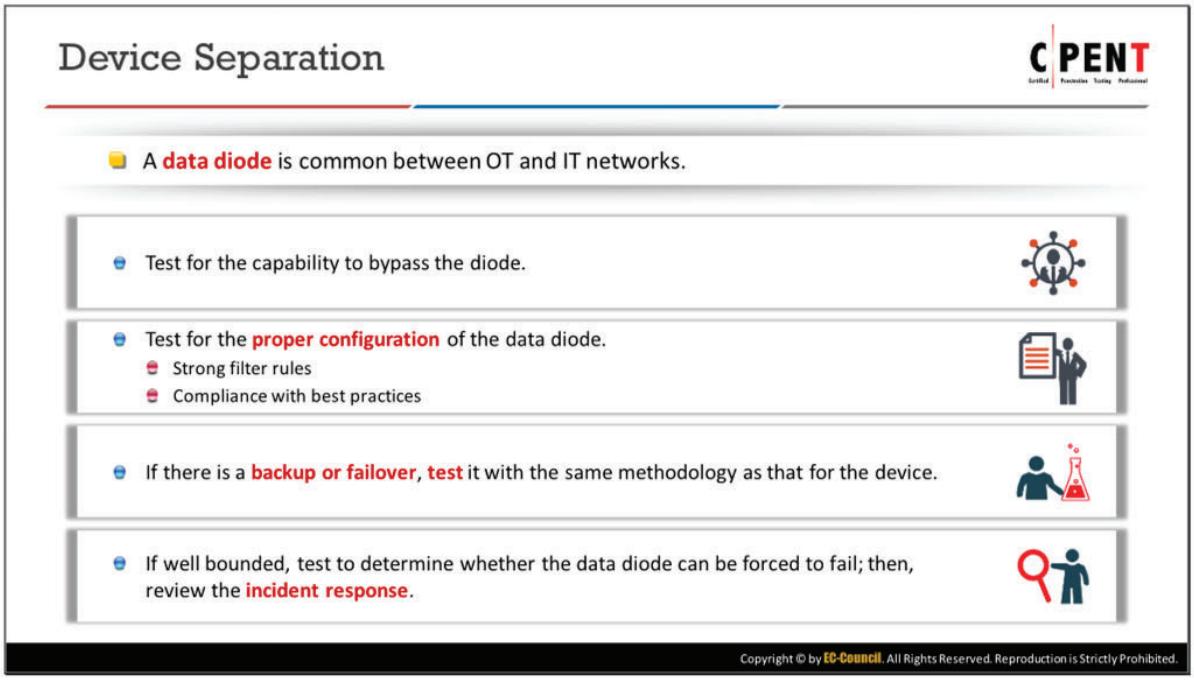
- None
- Authenticated as a normal user
- Administrator or super user access



Example Nessus Scan

Nessus scans decrease risk by removing TCP/UDP port scans and vulnerability probes. To use Nessus audit checks, the tester must create a new scan profile and disable all Nessus TCP, SYN, UDP, and SNMP port scans. Netstat and Ping port scans can be left open, but all Nessus plugins must be disabled, except for Windows/Linux compliance checks. After creating a new scan, it should be configured to use the new profile. As some of the included profiles might be dated, they should be thoroughly reviewed. However, Nessus constantly updates its profiles. Therefore, the tool must be kept updated, and all policies must be reviewed after an update to determine any changes.

Because Nessus is not free for commercial use, an open-source alternative called OpenVAS, which is actually a fork of Nessus, was started by the creator of Nessus. However, note that OpenVAS's code is in a continuously changing state, and its scans are much slower than those of Nessus. Nevertheless, it does work and is free.



Device Separation

Owing to some well-publicized attacks, some organizations deploy a data diode. Testers must be aware of this and must research the testing plan once these are installed because they require extensive testing. Both the Ukrainian Power Grid compromise and other attacks would have been prevented with a **well-configured** deterministic data diode. Note the emphasis on well configured: the configuration must be tested when performing penetration tests.

Implementation of Deterministic Unidirectional Devices

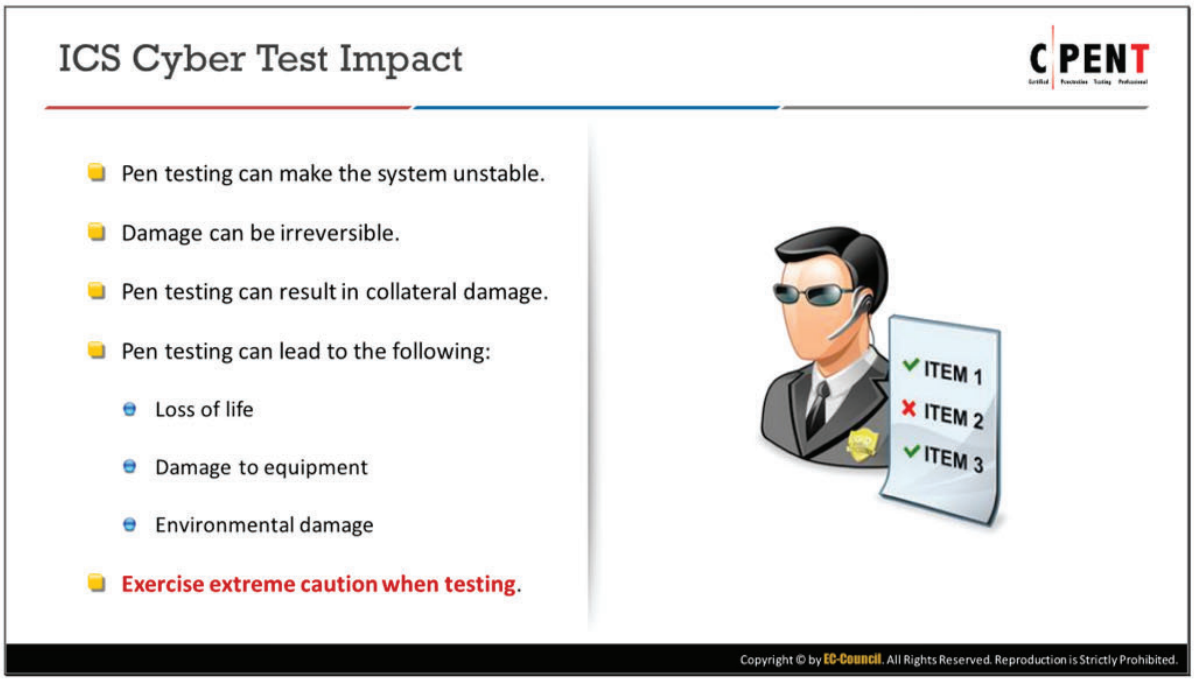
In any organization, separating the business network from the operational network should be of high priority, especially for critical infrastructure entities. Installing a data diode between IT and OT networks is an effective solution to thwarting cyber-attacks seeking to utilize a hard-wired network threat vector to target ICSs for exploitation. A properly installed and configured data diode offers a deterministic unidirectional solution to protect OT systems and equipment. However, it is important to note that a data diode is not the solution to all cyber-attack vectors. Documented cyber-attacks, originating from nearly anywhere in the world, have made use of multiple threat vectors to accomplish their goals.

A properly installed and configured data diode effectually protects upstream networks by allowing data communications to flow in one direction only. Functionally, data diodes may require special configurations in a TCP/IP environment where both send and receive packets are required. Many industries are not eager to employ deterministic unidirectional devices owing to the annulment of remote connectivity, specifically affecting vendors and after-hours support. With cyber-attacks continuing to increase in sophistication, the definitive protection of OT networks from exploitation should become a business requirement instead of a recommendation. The risk appetite of an organization usually affects whether unidirectional communication is acceptable.

The Nuclear Reactors, Materials, and Waste Sector of the critical infrastructure of the United States includes a regulatory requirement for unidirectional data flow between the IT and OT networks of commercial nuclear reactors.

As a result, during testing, there is a chance of encountering a data diode, which is not encountered in IT testing. Therefore, before testing any network with this type of device between IT and OT networks, data diodes must be thoroughly researched.

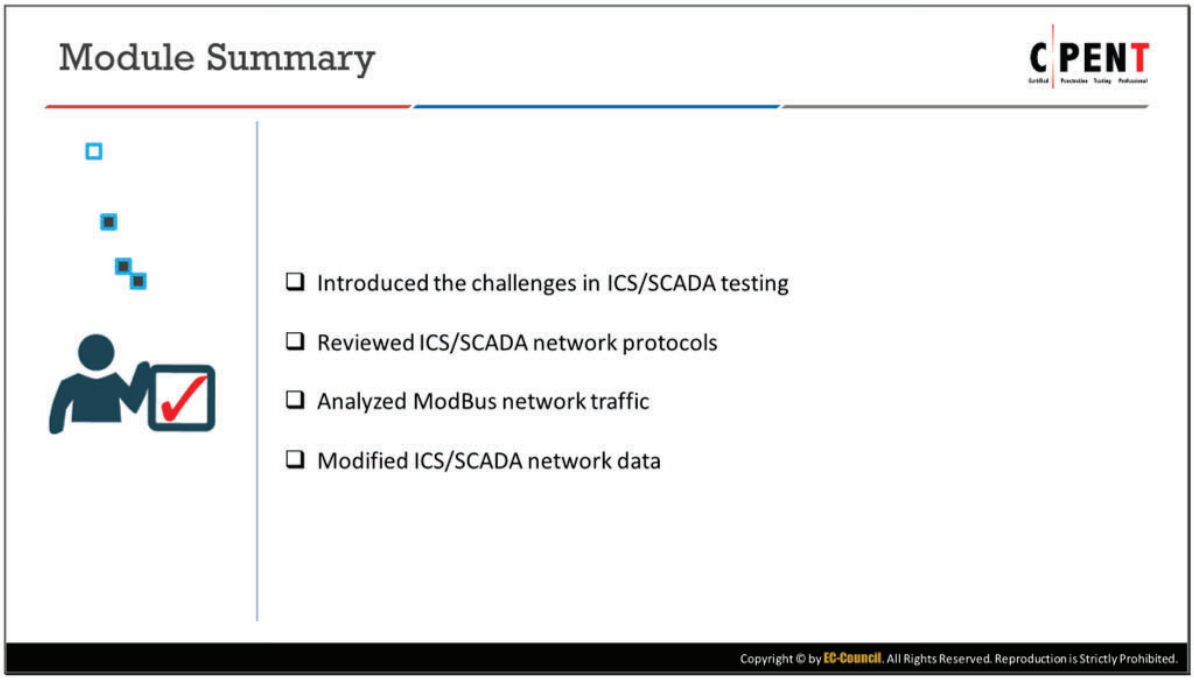
Pen testing a data diode from the IT side of the network should confirm that the diode is installed and configured appropriately to block communications that could be attempting to access OT equipment through the wired network. The architecture of the diode should be examined ahead of pen testing. If the data diode utilizes fiber optics, pen testing efforts would be voided. Additionally, the identification of bypasses around the diode requires different tactics (e.g., network sniffing) that should be performed in addition to pen testing.



ICS Cyber Test Impact

ICSs are an integral part of critical infrastructure, helping facilitate operations in vital sectors such as energy, oil and gas, water, transportation, and chemical manufacturing. The growing issue of cyber security and its impact on ICSs highlight the fundamental risks to a nation's critical infrastructure. Efficiently addressing ICS cyber security issues requires a clear understanding of current security challenges and specific defensive countermeasures. A professional tester must exercise caution when testing in an IT network, but an OT network requires even more due diligence before performing any testing. The tester must always proceed with caution and crawl to conclusions.

LAB SCADA Pen Testing



Module Summary

- Introduced the challenges of ICS/SCADA testing
- Reviewed the ICS/SCADA network protocols
- Analyzed ModBus network traffic
- Modified the ICS/SCADA network data