Blue\_Team\_Course Incident response notes:

Incident**response is the methodology an organization uses to respond to and manage a cyber attack.**

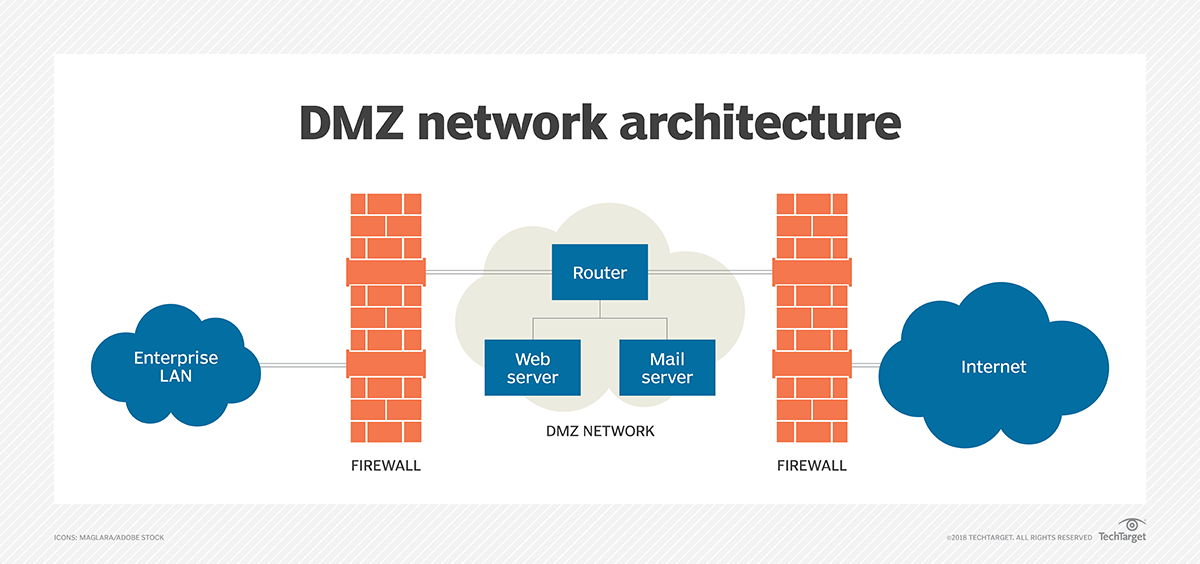
Incident response is a reactive approach and is closely aligned with disaster recovery efforts. Responding to these events in an organized manner with the right resources can save the business money by reducing recovery time and costs.

**All security incidents are security events, but not all events become incidents. A Security event is anything that *could* have a security implication, such as causing damage or disruption. Security incidents are security events that have resulted in damage to the organization.**

**Many organizations have procedures for handling computer-security incidents, and this is known as having an Incident Response Plan (IRP).  Many of these plans are based on the NIST SP 800 61r2 guidelines.**

Full **NIST** publication: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf>

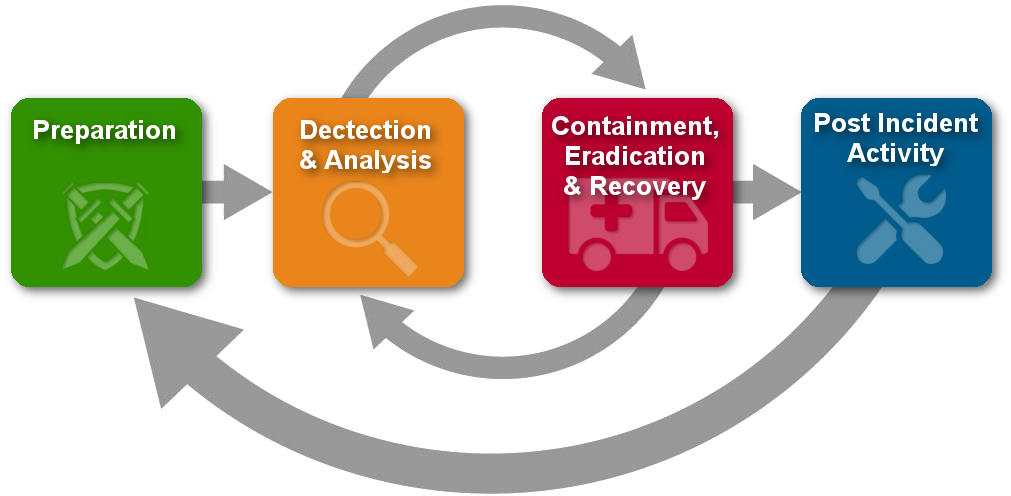
**Important Terms**:



**Group Policies:**

A Group Policy Object is a collection of settings systems administrators create with the Microsoft Management Console (MMC) Group Policy Editor. The GPO can be associated with one or more of the Active Directory containers, such as sites, domains, or organizational units(OUs).

**Incident Response Life Cycle**



Further reading: <https://cybersecurity.att.com/resource-center/ebook/insider-guide-to-incident-response/incident-response-tools>

**Basic Wireshark queries & filters:**

**For example, tcp.port == 80 displays packets that have a source or destination port of 80 (HTTP), and tcp.window\_size\_value >= 8000 displays TCP packets with a window size of 8000 bytes or over. Multiple filter statements can be chained by using logical operators, including and (&&) and (or/||). For example, to display TCP packets addressed to 192.168.1.7, you can use ip.dst\_host == 192.168.1.7 && tcp, and to display either NTP traffic or UDP traffic from/to port 20000, you can use ntp or udp.port == 20000. The not (!) operator excludes specific packets from being displayed, such as not ftp to exclude FTP packets from being displayed in the packet list. Finally, brackets can be used to group statements together.**

**Windows Management Instrumentation:**

**Windows Management Instrumentation (WMI) is an administration feature that facilitates the management of devices and applications in a network from a Windows system. WMI provides users with information about the status of local or remote computer systems and can be used to remotely execute code on other systems. It uses the WMI service for local and remote access and the**[**Server Message Block**](https://en.wikipedia.org/wiki/Server_Message_Block)**(SMB) and [Remote Procedure Call Service](https://en.wikipedia.org/wiki/Remote_procedure_call) (RPCS). MITRE states that organisations should monitor for WMI usage, which we can do using System Monitoring (Sysmon) from Sysinternals. We could then monitor the following event IDs:**

* [**Sysmon Event ID 19**](https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventid=90019)**–WmiEventFilter activity detected**
* [**Sysmon Event ID 20**](https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventid=90020)**–WmiEventConsumer activity detected**
* [**Sysmon Event ID 21**](https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventid=90021)**–WmiEventConsumerToFilter activity detected**

[**LSASS Memory – T1003.1:**](https://attack.mitre.org/techniques/T1003/001/)

**Local Security Authority Subsystem Service (LSASS). When a user logs onto a Windows system their credentials are stored in LSASS process memory which can be accessed by an Administrator or a SYSTEM-level user. If the adversary has control of an admin account they are able to dump everything from the LSASS memory and then brute force the password hashes offline to retrieve the plaintext versions. One can use credential dumping programs such as Mimikatz to do this.**

**Useful CMD & PowerShell commands:**

*wmic process get description, executablepath***- This command will display running processes and the associated binary file that was executed to create the process.**

*net user -*prints a list of all system users

*net localgroup administrators* - prints administrators(or whatever specified group)

*sc query | more* - lists all services and details about them.

netstat -ab  - prints open ports

**PowerShell:**

**Get-NetIPConfiguration and Get-NetIPAddress***- network related info*

**Get-LocalUser - Get-LocalUser -Name BTLO | select \****- we can use a specific username and list every property with the wildcard \**

**Get-Service | Where Status -eq "Running" | Out-GridView***- The above command let’s us quickly identify running services on the system. By piping ( | ) the command to Out-GridView, we are telling PowerShell to show us the results in a nice windows, which is much easier to work with than outputting the results to the PowerShell window.*

**Get-Process | Format-Table View priority***- Another great command is the ability to group running processes by their priority value. Using the above command we can see the process name, the process ID (PID), and other information, where different priority ratings are grouped into tables.*

**Get-Process -Id 'idhere' | Select \*** *- We can collect specific information from a service by including the name in the command (-Name ‘namehere’) or the Id, as shown above and below. Piping to Select \* provides us with all the properties.*

**Get-ScheduledTask -***Similar to Services, Scheduled Tasks are often abused and utilized a common persistence technique. With the above command we can list tasks that are set to run after certain conditions are met.*

**DeepBlueCLI:**

Can identify a wide range of attacks provided with relevant Cringedows Event logs and/or Sysmon logs.

To analyze real-time local sys logs we'd use:

**./DeepBlue.ps1 -log security .\path\to\an\.evtx**

**./DeepBlue.ps1 -log system**

Short guide: [DeepBlueCLI (logs Powershell, free) (cyberhuntingguide.net)](https://www.cyberhuntingguide.net/deepbluecli-logs-powershell-free.html" \l ":~:text=DeepBlueCLI%20is%20a%20tool%20to%20log%20PowerShell%20events.,attacker%20can%20use%20these%20tools%20without%20being%20captured.)