CIT 184 Computer Security Incident Response

EPIC
READY SET LEARN

Computer security incident response has become an important component of information technology (IT) programs. This lesson provides concepts and techniques for proper incident handling and documentation as part of an incident response team.

The content of this lesson is derived from the National Institute of Standards and Technology Computer Security Incident Handling Guide (Special Publication 800-61 Revision 2) and Guide to Malware Incident Prevention and Handling for Desktops and Laptops (Special Publication 800-83 Revision 1).

- Need for Incident Response
- Preparation
- Detection & Analysis
- Containment, Eradication & Recovery

Post-Incident Activity

Check Your Understanding (Sorting Activity)

Quiz

This workforce product was funded by a grant awarded by the U.S. Department of Labor's Employment and Training Administration. The product was created by the grantee and does not necessarily reflect the official position of the U.S. Department of Labor. The Department of Labor makes no guarantees, warranties, or assurances of any kind, express or implied, with respect to such information, including any information on linked sites and including, but not limited to, accuracy of the information or its completeness, timeliness, usefulness, adequacy, continued availability, or ownership.

Need for Incident Response

The Need for Incident Response

Computer security incident response has become an important component of information technology (IT) programs. New types of security-related incidents emerge frequently. Preventive activities based on the results of risk assessments can lower the number of incidents, but not all incidents can be prevented. An incident response capability is therefore

necessary for rapidly detecting incidents, minimizing loss and destruction, mitigating the weaknesses that were exploited, and restoring IT services.

The number-one incident preparation and prevention strategy is organizational policy.

- Organizations must create, provision, and operate a formal incident response capability. Federal law requires Federal agencies to report incidents to the United States Computer Emergency Readiness Team (US-CERT) office within the Department of Homeland Security (DHS).
- Organizations should reduce the frequency of incidents by effectively securing networks, systems, and applications.
- Organizations should document their guidelines for interactions with other organizations regarding incidents.
- Organizations should be generally prepared to handle any incident but should focus on being prepared to handle incidents that use common attack vectors.
- Organizations should emphasize the importance of incident detection and analysis throughout the organization.
- Organizations should create written guidelines for prioritizing incidents.
- Organizations should use the lessons learned process to gain value from incidents.

One of the first considerations should be to create an organization-specific definition of the term “incident” so that the scope of the term is clear. The organization should decide what

services the incident response team should provide, consider which team structures and models can provide those services, and select and implement one or more incident response teams. Incident response plan, policy, and procedure creation is an important part of establishing a team, so that incident response is performed effectively, efficiently, and consistently, and so that the team is empowered to do what needs to be done. The plan, policies, and procedures should reflect the team's interactions with other teams within the organization as well as with outside parties, such as law enforcement, the media, and other incident response organizations.

Events and Incidents

An *event* is any observable occurrence in a system or network. Events include a user connecting to a file share, a server receiving a request for a web page, a user sending email, and a firewall blocking a connection attempt. *Adverse events* are events with a negative consequence, such as system crashes, packet floods, unauthorized use of system privileges, unauthorized access to sensitive data, and execution of malware that destroys data. This guide addresses only adverse events that are computer security-related, not those caused by natural disasters, power failures, etc.

A *computer security incident* is a violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices. Examples of incidents are:

- An attacker commands a botnet to send high volumes of connection requests to a web server, causing it to crash.
- Users are tricked into opening a "quarterly report" sent via email that is actually malware; running the tool has infected their computers and established connections with an external host.
- An attacker obtains sensitive data and threatens that the details will be released publicly if the organization does not pay a designated sum of money.
- A user provides or exposes sensitive information to others through peer-to-peer file sharing services.

Need for Incident Response

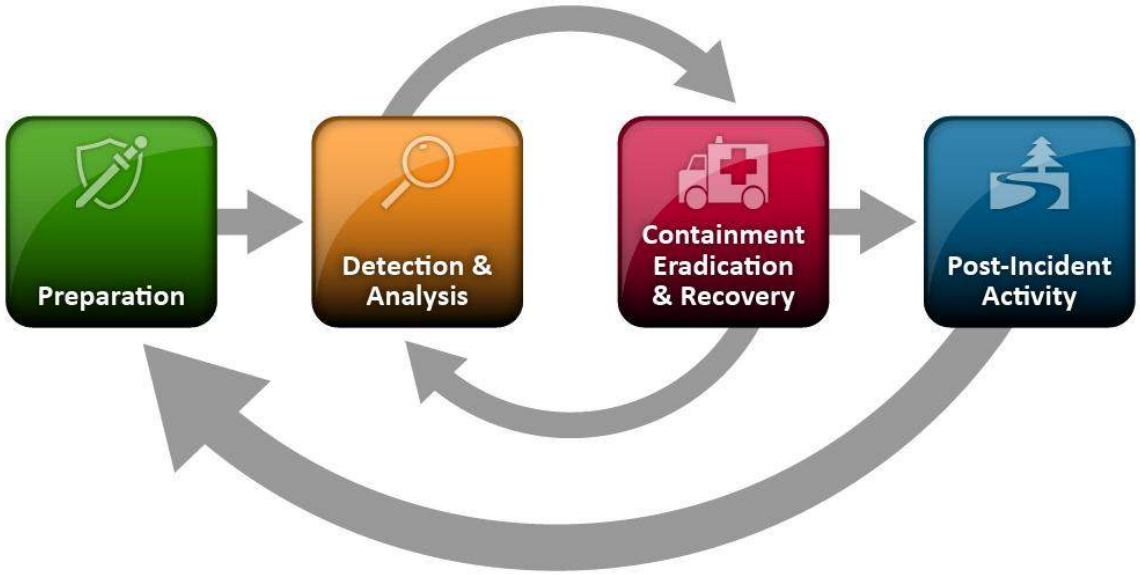
Attacks frequently compromise personal and business data, and it is critical to respond quickly and effectively when security breaches occur. The concept of computer security incident response has become widely accepted and implemented. One of the benefits of having an incident response capability is that it supports responding to incidents systematically (i.e., following a consistent incident handling methodology) so that the appropriate actions are taken. Incident response helps personnel to minimize loss or theft of information and disruption of services caused by incidents. Another benefit of incident response is the ability to use information gained during incident handling to better prepare for handling future incidents and to provide stronger protection for systems and data. An incident response capability also helps with dealing properly with legal issues that may arise during incidents.

Predefining incident responses enables the organization to react to a detected incident quickly and effectively, without confusion or wasted time and effort.

Incident Response Life Cycle and Malware Incidents

The following lessons focus on the aspects of incident handling that are specific to malware incidents. Each phase of the incident response life cycle will be briefly described. Although malware incidents are the focus, similar techniques and the incident response life cycle can be utilized for various types of computer security incident.

Preparation



Incident Response Life Cycle

The incident response process has several phases. The next sections briefly describes the major phases of the incident response process specific to malware incidents —preparation, detection and analysis, containment, eradication and recovery, and post-incident activity. The figure above illustrates the incident response life cycle.

Preparation

The initial phase of malware incident response involves performing preparatory activities, such as developing malware-specific incident handling procedures and training programs for incident response teams. As described in Section 3, the preparation phase also involves using policy, awareness activities, vulnerability mitigation, and security tools to reduce the number of malware incidents. Despite these measures, residual risk will inevitably persist, and no solution is foolproof. Detection of malware infections is thus necessary to alert the organization whenever incidents occur. Early detection is particularly important for malware incidents because they are more likely than other types of incidents to increase their impact over time, so faster detection and handling can help reduce the number of infected hosts and the damage done.

Organizations should perform preparatory measures to ensure that they are capable of responding effectively to malware incidents. Sections below describe several recommended preparatory measures, including building and maintaining malware-related skills within the incident response team, facilitating communication and coordination throughout the organization, and acquiring necessary tools and resources.

Building and Maintaining Malware-Related Skills

All malware incident handlers should have a solid understanding of how each major category of malware infects hosts and spreads. Also, incident handlers should be familiar with the organization's implementations and configurations of malware detection tools so that they are better able to analyze supporting data and identify the characteristics of threats. Incident handlers doing in-depth malware analysis should have strong skills in that area and be familiar with the numerous tools for malware analysis.

Malware incident handlers should keep abreast of the ever-evolving landscape of malware threats and technology. Besides conducting malware-related training and exercises, organizations should also seek other ways of building and maintaining skills. One possibility is to have incident handlers temporarily work as antivirus engineers or administrators so that they can gain new technical skills and become more familiar with antivirus staff procedures and practices.

Facilitating Communication and Coordination

One of the most common problems during malware incident handling is poor communication and coordination. Anyone involved in an incident, including users, can inadvertently cause additional problems because of a limited view or understanding of the situation. To improve communication and coordination, an organization should designate in advance a few individuals or a small team to be responsible for coordinating the organization's responses to malware incidents. The coordinator's primary goal is to maintain situational awareness by gathering all pertinent information, making decisions that are in the best interests of the organization, and communicating pertinent information and decisions to all relevant parties in a timely manner. For malware incidents, the relevant parties often include end users, who might be given instructions on how to avoid infecting their hosts, how to recognize the signs of an infection, and what to do if a host appears to be infected. The coordinator also needs to provide technical guidance and instructions to all staff assisting with containment, eradication, and recovery efforts, as well as giving management regular updates on the status of the response and the current and likely future impact of the incident. Another possible role for the coordinator is interacting with external parties, such as other incident response teams facing similar malware issues.

Organizations should also establish a point of contact for answering questions about the legitimacy of malware alerts. Many organizations use the IT help desk as the initial point of contact and give help desk agents access to sources of information on real malware threats and virus hoaxes so that they can quickly determine the legitimacy of an alert and provide users with guidance on what to do. Organizations should caution users not to forward malware alerts to others without first confirming that the alerts are legitimate.

Organizations should implement several incident reporting mechanisms, such as phone numbers, email addresses, online forms, and secure instant messaging systems that users can use to report suspected incidents. At least one mechanism should permit people to report incidents anonymously.

Acquiring Tools and Resources

Organizations should also ensure that they have the necessary tools (hardware and software) and resources to assist in malware incident handling. Many incident response teams create a jump kit, which is a portable case that contains materials that may be needed during an investigation. The jump kit should be ready to go at all times. Each jump kit typically includes a laptop, loaded with appropriate software (e.g., packet sniffers, digital forensics). Other important items include backup devices, blank media, and basic networking equipment and cables. Because the purpose of having a jump kit is to facilitate faster responses, the team should avoid borrowing items form the jump kit.

Preventing Incidents

Keeping the number of incidents reasonably low is very important to protect the business processes of the organization. If security controls are insufficient, higher volumes of incidents may occur, overwhelming the incident response team. This can lead to slow and incomplete responses, which translate to a larger negative business impact (e.g., more extensive damage, longer periods of service and data unavailability).

An incident response team may be able to identify problems that the organization is otherwise not aware of; the team can play a key role in risk assessment and training by identifying gaps. The following text provides a brief overview of some of the main recommended practices for securing networks, systems, and applications:

- **Risk Assessments.** Periodic risk assessments of systems and applications should determine what risks are posed by combinations of threats and vulnerabilities. This should include understanding the applicable threats, including organization-specific threats. Each risk should be prioritized, and the risks can be mitigated, transferred, or accepted until a reasonable overall level of risk is reached. Another benefit of conducting risk assessments regularly is that critical resources are identified, allowing staff to emphasize monitoring and response activities for those resources.

- **Host Security.** All hosts should be hardened appropriately using standard configurations. In addition to keeping each host properly patched, hosts should be configured to follow the principle of least privilege—granting users only the privileges necessary for performing their authorized tasks. Hosts should have

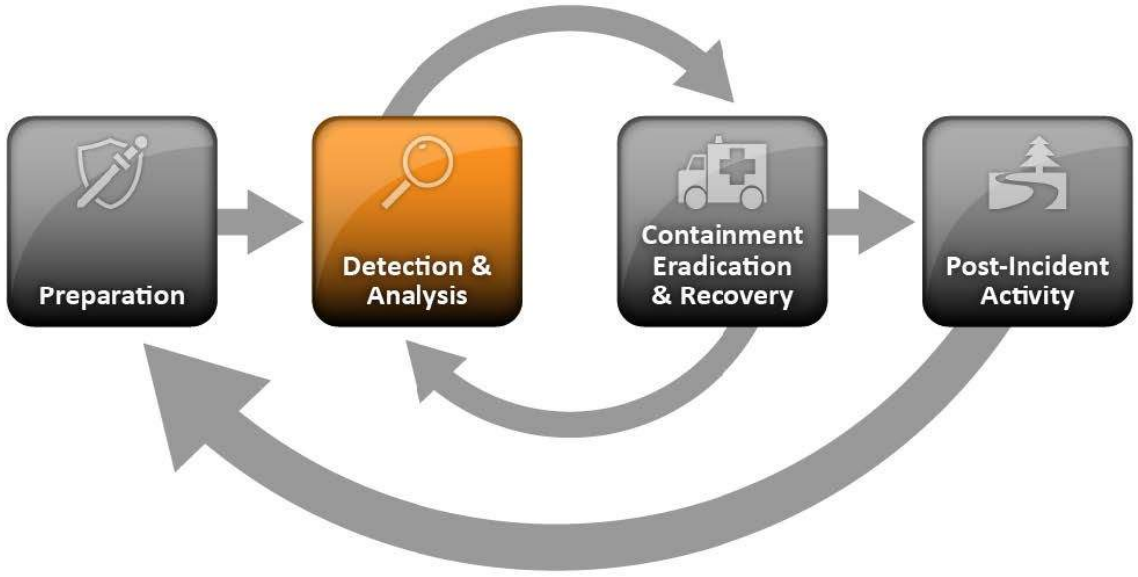
auditing enabled and should log significant security-related events. The security of hosts and their configurations should be continuously monitored. Many organizations use Security Content Automation Protocol (SCAP) expressed operating system and application configuration checklists to assist in securing hosts consistently and effectively.

- **Network Security.** The network perimeter should be configured to deny all activity that is not expressly permitted. This includes securing all connection points, such as virtual private networks (VPNs) and dedicated connections to other organizations.

- **Malware Prevention.** Software to detect and stop malware should be deployed throughout the organization. Malware protection should be deployed at the host level (e.g., server and workstation operating systems), the application server level (e.g., email server, web proxies), and the application client level (e.g., email clients, instant messaging clients).

- **User Awareness and Training.** Users should be made aware of policies and procedures regarding appropriate use of networks, systems, and applications. Applicable lessons learned from previous incidents should also be shared with users so they can see how their actions could affect the organization. Improving user awareness regarding incidents should reduce the frequency of incidents. IT staff should be trained so that they can maintain their networks, systems, and applications in accordance with the organization's security standards.

Detection & Analysis



Incident Response Life Cycle (Detection and Analysis

Detection

For many organizations, the most challenging part of the incident response process is accurately detecting and assessing possible incidents-determining whether an incident has occurred and, if so, the type, extent, and magnitude of the problem. Signs of an incident fall into one of two categories: precursors and indicators. A precursor is a sign that an incident may occur in the future. An indicator is a sign that an incident may have occurred or may be occurring now..

Organizations should strive to detect and validate malware incidents rapidly to minimize the number of infected hosts and the amount of damage the organization sustains. Because malware can take many forms and be distributed through many means, there are many possible signs of a malware incident and many locations within an organization where the signs might be recorded or observed. It sometimes takes considerable analysis, requiring extensive technical knowledge and experience, to confirm that an incident has been caused by malware, particularly if the malware threat is new and unknown. After malware incident detection and validation, incident handlers should determine the type, extent, and magnitude of the problem as quickly as possible so that the response to the incident can be given the appropriate priority. The following lesson provides guidance on identifying the characteristics of incidents, identifying infected hosts, prioritizing incident response efforts, and analyzing malware, respectively.

Identifying Malware Incident Characteristics

Because no indicator is completely reliable—even antivirus software might miscategorize benign activity as malicious—incident handlers need to analyze any suspected malware incident and validate that malware is the cause. In some cases, such as a massive, organization-wide infection, validation may be unnecessary because the nature of the incident is obvious. The goal is for incident handlers to be as certain as feasible that an incident is caused by malware and to have a basic understanding of the type of malware threat responsible, such as a worm or a Trojan horse. If the source of the incident cannot easily be confirmed, it is often better to respond as if it were caused by malware and to alter response efforts if it is later determined that malware is not involved. Waiting for conclusive evidence of malware might have a serious negative impact on response efforts and significantly increase the damage sustained by the organization.

As part of the analysis and validation process, incident handlers typically identify characteristics of the malware activity by examining detection sources. Understanding the activity's characteristics is very helpful in assigning an appropriate priority to the incident response efforts and planning effective containment, eradication, and recovery activities. Incident handlers should collaborate with security administrators in advance to identify data sources that can aid in detecting malware information and to understand what types of information each data source may record. In addition to the obvious sources of data, such as antivirus software, intrusion detection system (IDS), and security information and event

management (SIEM) technologies, incident handlers should be aware of and use secondary sources as appropriate.

Once incident handlers have reviewed detection source data and identified characteristics of the malware, the handlers could search for those characteristics in antivirus vendors' malware databases and identify which instance of malware is the most likely cause. If the malware has been known for some time, it is likely that antivirus vendors will have a substantial amount of information on it, such as the following:

- Malware category (e.g., virus, worm, Trojan horse)
- Services, ports, protocols, etc. that are attacked
- Vulnerabilities that are exploited (e.g., software flaws, misconfigurations, social engineering)
- Malicious filenames, sizes, content, and other metadata (e.g., email subjects, web URLs)
- Which versions of operating systems, devices, applications, etc., may be affected
- How the malware affects the infected host, including the names and locations of affected files, altered configuration settings, installed backdoor ports, etc.
- How the malware propagates and how to approach containment
- How to remove the malware from the host.

Unfortunately, the newest threats might not be included in malware databases for several hours or days, depending on the relative importance of the threat, and highly customized threats might not be included in malware databases at all. Therefore, incident handlers may need to consult other sources of information. One option is using public security mailing lists, which might contain first-hand accounts of malware incidents; however, such reports are

often incomplete or inaccurate, so incident handlers should validate any information obtained from these sources. Another potentially valuable source of malware characteristic information is peers at other organizations. Other organizations may have already been affected and gathered data on the threat. Establishing and maintaining good relationships with peers at other organizations that face similar problems can be advantageous for all involved. An alternative source of information is self-discovery by performing malware analysis. This is particularly important if the malware is highly customized; there may be no other way of getting details for the malware other than doing a hands-on analysis.

Identifying Infected Hosts

Identifying hosts that are infected by malware is part of every malware incident. Once identified, infected hosts can undergo the appropriate containment, eradication, and recovery actions. Unfortunately, identifying all infected hosts is often complicated by the dynamic nature of computing. For instance, people shut hosts down, disconnect them from networks, or move them from place to place, making it extremely difficult to identify which hosts are currently infected. In addition, some hosts can boot to multiple OSs or use virtual operating system software; an infection in one OS instantiation might not be detectable when a host is currently using another OS.

Given the number of malware threats, all infection identification should be performed through automated means. Manual identification methods, such as relying on users to identify and report infected hosts, and having technical staff personally check each host, are not feasible for most situations. Organizations should carefully consider host identification issues so that they are prepared to use multiple identification strategies as part of implementing effective containment strategies. Organizations should also determine which types of identifying information might be needed and what data sources might record the information. For example, a host's current IP address is typically needed for remote actions; of course, a host's physical location is needed for local actions. One piece of information can often be used to determine others, such as mapping an IP address to a media access control (MAC) address, which could then be mapped to a switch serving a particular group of offices. If an IP address can be mapped to a host owner or user—for example, by recording the mapping during network login—the owner or user can be contacted to provide the host's location.

The difficulty in identifying the physical location of an infected host depends on several factors. In a managed environment, identifying a host's location is often relatively easy because of the standardized manner in which things are done. For example, host names might contain the user's ID or office number, or the host's serial number (which can be tied to a user ID). Also, asset inventory management tools might contain current information on host characteristics. In other environments, especially those in which users have full control over their hosts and network management is not centralized, it might be challenging to link a machine to a location. For example, an administrator might know that the host at address 10.3.1.70 appears to be infected but not have any idea where that machine resides or who uses it. Administrators might need to track down an infected host through network devices. For example, a switch port mapper can poll switches for a particular IP address and identify the switch port number and host name associated with that IP address. If the infected host is several switches away, it can take hours to track down a single machine; if the infected host is not directly switched, the administrator might still need to manually trace connectivity through various wiring closets and network devices. An alternative is to pull the network cable or shut down the switch port for an apparently infected host and wait for a user to report an outage. This approach can inadvertently cause a loss of connectivity for small numbers of uninfected hosts, but if performed carefully as a last-resort identification and containment method, it can be quite effective.

Some organizations first make reasonable efforts to identify infected hosts and perform containment, eradication, and recovery efforts on them, then implement measures to prevent hosts that have not been verified as uninfected and properly secured from attaching to the network. These measures should be discussed well in advance, and incident handlers should have prior written permission to lock out hosts under certain circumstances. Generally, lockout measures are based on the characteristics of particular hosts, such as MAC addresses or static IP addresses, but lockouts can also be performed based on user ID if a host is associated with a single user. Another possibility is to use network login scripts to identify and deny access to infected hosts, but this might be ineffective if an infected host starts spreading malware after system boot but before user authentication. Having a separate VLAN for infected or unverified hosts can provide a good way to lock out hosts, as long as the mechanism to detect infections is reliable. Although lockout methods might be needed only under extreme circumstances, organizations should think in advance about how individual hosts or users could be locked out so that if needed, lockouts can be performed rapidly.

Incident Documentation

An incident response team that suspects that an incident has occurred should immediately start recording all facts regarding the incident. A logbook is an effective and simple medium for this, but laptops, audio recorders, and digital cameras can also serve this purpose. Documenting system events, conversations, and observed changes in files can lead to a more efficient, more systematic, and less error-prone handling of the problem. Every step taken from the time the incident was detected to its final resolution should be documented and timestamped. Every document regarding the incident should be dated and signed by the incident handler. Information of this nature can also be used as evidence in a court of law if legal prosecution is pursued. The incident response team should maintain records about the status of incidents, along with other pertinent information. Using an application or a database, such as an issue tracking system, helps ensure that incidents are handled and resolved in a timely manner. The issue tracking system should contain information on the following:

- The current status of the incident (new, in progress, forwarded for investigation, resolved, etc.)
- A summary of the incident
- Indicators related to the incident
- Other incidents related to this incident
- Actions taken by all incident handlers on this incident
- Chain of custody, if applicable
- Impact assessments related to the incident
- Contact information for other involved parties (e.g., system owners, system administrators)
- A list of evidence gathered during the incident investigation

- Comments from incident handlers
- Next steps to be taken (e.g., rebuild the host, upgrade an application).

The incident response team should safeguard incident data and restrict access to it because it often contains sensitive information—for example, data on exploited vulnerabilities, recent security breaches, and users that may have performed inappropriate actions. For example, only authorized personnel should have access to the incident database. Incident communications (e.g., emails) and documents should be encrypted or otherwise protected so that only authorized personnel can read them.

Prioritizing Incident Response

Once a malware incident has been validated, the next activity is to prioritize its handling. Certain forms of malware, such as worms, tend to spread very quickly and can cause a substantial impact in minutes or hours, so they often necessitate a high-priority response. Other forms of malware, such as Trojan horses, tend to affect a single host; the response to such incidents should be based on the value of the data and services provided by the host. Organizations should establish a set of criteria that identify the appropriate level of response for various malware-related situations. The criteria should incorporate considerations such as the following:

- How the malware entered the environment and what transmission mechanisms it uses
- What type of malware it is (e.g., virus, worm, Trojan horse)
- Which types of attacker tools are placed onto the host by the malware
- What networks and hosts the malware is affecting and how it is affecting them
- How the impact of the incident is likely to increase in the following minutes, hours, and days if the incident is not contained.

Malware Analysis

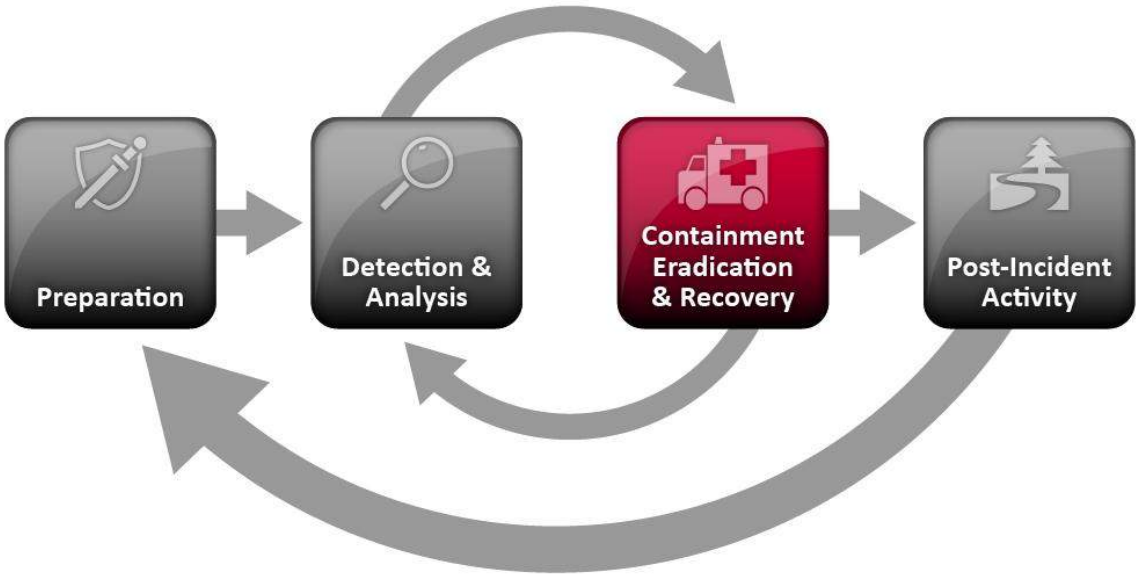
Incident handlers can study the behavior of malware by analyzing it either actively (executing the malware) or forensically (examining the infected host for evidence of malware). Forensic approaches are safer to perform on an infected host because they can examine the host without allowing the malware to continue executing. However, sometimes it is significantly faster and easier to analyze malware by monitoring it during execution. Such active approaches are best performed on malware test systems instead of production hosts, to minimize possible damage caused by allowing the malware to execute.

Ideal active approaches involve an incident handler acquiring a malware sample from an infected host and placing the malware on an isolated test system. Test systems often have a virtualized OS image; copies of these builds can be infected, isolating any infection within the virtualized OS, and the infected image can be replaced with a known good image after the analysis is complete. On such test systems, the host OS is kept uninfected so it can be used to monitor the execution of the malware within the virtualized OS. The test system should include up-to-date tools for identifying malware (e.g., antivirus software, intrusion detection systems), listing the currently running processes, and displaying network connections, as well as many other potentially helpful utilities. There are various websites and books that provide detailed instructions on setting up malware test systems and their tools; further discussion of them is outside the scope of this publication. Malware test systems are helpful not only for analyzing current malware threats without the risk of inadvertently causing additional damage to the organization, but also for training staff in malware incident handling.

Forensic approaches involve booting a forensic environment and using it to study the stored information from an infected host. The toolsets for forensic analysis greatly overlap those for active analysis; similarly, there are various websites and books available that explain how to create forensic analysis environments. There are two basic approaches: create a bootable forensic environment on write-protected removable media, or use a forensic workstation and connect it to the storage of the infected host (e.g., hard drive). The motivation for using such a trusted toolkit instead of relying on the information reported by the infected host's OS is that malware on the host may have disabled or altered the functionality of the security tools

on the infected host, such as antivirus software, so that they do not report malicious activity. By running tools from a protected, verified toolkit, incident handlers can gain a more accurate understanding of the activity on the host.

Containment, Eradication & Recovery



Incident Response Life Cycle (Containment, Eradication & Recovery)

Containment

In containing a malware incident, it is also important to understand that stopping the spread of malware does not necessarily prevent further damage to hosts. Malware on a host might continue to exfiltrate sensitive data, replace OS files, or cause other damage. In addition, some instances of malware are designed to cause additional damage when network connectivity is lost or other containment measures are performed. For example, an infected host might run a malicious process that contacts another host periodically. If that connectivity is lost because the infected host is disconnected from the network, the malware

might overwrite all the data on the host's hard drive. For these reasons, handlers should not assume that just because a host has been disconnected from the network, further damage to the host has been prevented, and in many cases, should begin eradication efforts as soon as possible to prevent more damage.

Organizations should have strategies and procedures in place for making containment-related decisions that reflect the level of risk acceptable to the organization. For example, an organization might decide that infected hosts performing critical functions should not be disconnected from networks or shut down if the likely damage to the organization from those functions being unavailable would be greater than the security risks posed by not isolating or shutting down the host. Containment strategies should support incident handlers in selecting the appropriate combination of containment methods based on the characteristics of a particular situation.

Containment methods can be divided into four basic categories: relying on user participation, performing automated detection, temporarily halting services, and blocking certain types of network connectivity.

Containment Through User Participation

At one time, user participation was a valuable part of containment efforts, particularly during large-scale incidents in non-managed environments. Users were provided with instructions on how to identify infections and what measures to take if a host was infected, such as calling the help desk, disconnecting the host from the network, or powering off the host. The instructions might also cover malware eradication, such as updating antivirus signatures and performing a host scan, or obtaining and running a specialized malware eradication utility. As hosts have increasingly become managed, user participation in containment has sharply decreased. However, having users perform containment actions is still helpful in non-managed environments and other situations in which use of fully automated containment methods is not feasible.

Although user participation can be very helpful for containment, organizations should not rely on this means for containing malware incidents unless absolutely necessary.

Containment Through Automated Detection

Many malware incidents can be contained primarily through the use of the automated technologies for preventing and detecting infections. These technologies include antivirus software, content filtering, and intrusion prevention software. Because antivirus software on hosts can detect and remove infections, it is often the preferred automated detection method for assisting in containment. However, as previously discussed, many of today's malware threats are novel, so antivirus software and other technologies often fail to recognize them as being malicious. Also, malware that compromises the OS may disable security controls such as antivirus software, particularly in unmanaged environments where users have greater control over their hosts. Containment through antivirus software is not as robust and effective as it used to be.

Organizations should be prepared to use other security tools to contain the malware until the antivirus signatures can perform the containment effectively, if antivirus signatures become available at all. Automated detection methods other than antivirus software may include:

- Content Filtering
- Network-Based IPS Software
- Executable Blacklisting.

Containment Through Disabling Services or Connectivity

Some malware incidents necessitate more drastic and potentially disruptive measures for containment. These incidents make extensive use of a particular service. Containing such an incident quickly and effectively might be accomplished through a loss of services, such as shutting down a service used by malware, blocking a certain service at the network perimeter, or disabling portions of a service (e.g., large mailing lists). Also, a service might provide a channel for infection or for transferring data from infected hosts—for example, a botnet command and control channel using Internet Relay Chat (IRC). In either case, shutting down the affected services might be the best way to contain the infection without losing all

services. This action is typically performed at the application level (e.g., disabling a service on servers) or at the network level (e.g., configuring firewalls to block IP addresses or ports associated with a service). The goal is to disable as little functionality as possible while containing the incident effectively. To support the disabling of network services, organizations should maintain lists of the services they use and the TCP and UDP ports used by each service.

Containing incidents by placing temporary restrictions on network connectivity can be very effective. For example, if infected hosts attempt to establish connections with an external host to download rootkits, handlers should consider blocking all access to the external host (by IP address or domain name, as appropriate). Similarly, if infected hosts within the organization attempt to spread their malware, the organization might block network traffic from the hosts' IP addresses to control the situation while the infected hosts are physically located and disinfected. An alternative to blocking network access for particular IP addresses is to disconnect the infected hosts from the network, which could be accomplished by reconfiguring network devices to deny network access or physically disconnecting network cables from infected hosts.

Containment Recommendations

Containment can be performed through many methods in the four categories described above (users, automated detection, loss of services, and loss of connectivity). Because no single malware containment category or individual method is appropriate or effective in every situation, incident handlers should select a combination of containment methods that is likely to be effective in containing the current incident while limiting damage to hosts and reducing the impact that containment methods might have on other hosts. For example, shutting down all network access might be very effective at stopping the spread of malware, but it would also allow infections on hosts to continue damaging files and would disrupt many important functions of the organization.

The most drastic containment methods can be tolerated by most organizations for only a brief period of time. Accordingly, organizations should support sound containment decisions by having policies that clearly state who has authority to make major containment decisions

and under what circumstances various actions (e.g., disconnecting subnets from the Internet) are appropriate.

Eradication

Although the primary goal of eradication is to remove malware from infected hosts, eradication is typically more involved than that. If an infection was successful because of a host vulnerability or other security weakness, such as an unsecured file share, then eradication includes the elimination or mitigation of that weakness, which should prevent the host from becoming reinfected or becoming infected by another instance of malware or a variant of the original threat. Eradication actions are often consolidated with containment efforts. For example, organizations might run a utility that identifies infected hosts, applies patches to remove vulnerabilities, and runs antivirus software that removes infections. Containment actions often limit eradication choices; for example, if an incident is contained by disconnecting infected hosts from the primary network, the hosts should either be connected to a separate VLAN so that they can be updated remotely, or patched and reconfigured manually. Because the hosts are disconnected from the primary network, the incident handlers will be under pressure to perform eradication actions on the hosts as quickly as possible so that the users can regain full use of their hosts.

Different situations necessitate various combinations of eradication techniques. In cases where disinfection is possible, the most common tools for eradication are antivirus software, vulnerability management technologies, network access control software, and other tools designed to remove malware and correct vulnerabilities. Automated eradication methods, such as triggering antivirus scans remotely, are much more efficient than manual methods, such as visiting infected hosts in person and running disinfection software from a CD. Some situations necessitate user participation in containment and eradication activities. Providing instructions and software updates to users works in some cases, but other users might need assistance. Having formal or informal walk-up help desk areas at major facilities can also be effective and is more efficient and convenient than having IT staff locate and interrupt each affected user. During major incidents, additional IT staff members can be relieved of other duties temporarily to assist in eradication efforts. For locations without IT staff, it is often helpful to have a few people trained in basic eradication actions so that they can take care of their own hosts. Organizations should be prepared to perform a few different types of eradication efforts simultaneously if needed.

For many malware incidents, simple disinfection is not feasible, so it is necessary to rebuild all infected hosts as part of eradication efforts. Rebuilding includes the reinstallation and securing of the OS and applications (or restoration of known good OS and application backups, including the use of built-in OS rollback capabilities), and the restoration of data from known good backups. Some types of malware are extremely difficult to remove from hosts; even if they can be removed, each host's OS may be damaged, possibly to the point where the hosts cannot boot. Rebuilding is also the best eradication option when the actions performed on an infected host are unknown.

In general, organizations should rebuild any host that has any of the following incident characteristics, instead of performing typical eradication actions (disinfection):

- One or more attackers gained administrator-level access to the host.
- Unauthorized administrator-level access to the host was available to anyone through a backdoor, an unprotected share created by a worm, or other means.
- System files were replaced by a Trojan horse, backdoor, rootkit, attacker tools, or other means.
- The host is unstable or does not function properly after the malware has been eradicated by antivirus software or other programs or techniques. This indicates that either the malware has not been eradicated completely or that it has caused damage to important system or application files or settings.
- There is doubt about the nature of and extent of the infection or any unauthorized access gained because of the infection.

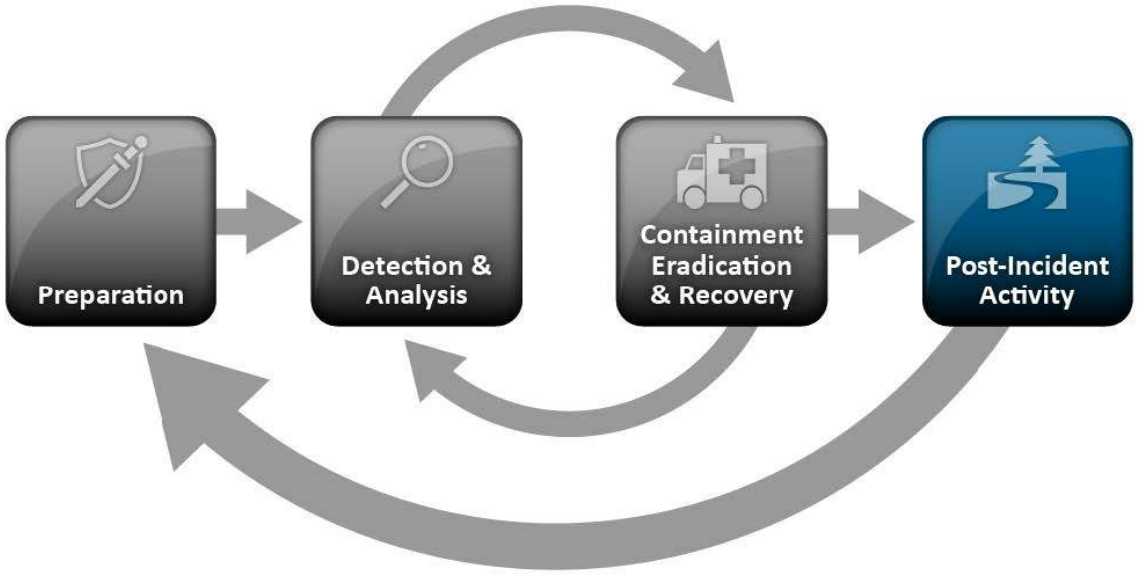
Recovery

The two main aspects of recovery from malware incidents are restoring the functionality and data of infected hosts and removing temporary containment measures. Additional actions to restore hosts are not necessary for most malware incidents that cause limited host damage (for example, an infection that simply altered a few data files and was completely removable

with antivirus software). For malware incidents that are far more damaging, such as Trojan horses, rootkits, or backdoors, corrupting thousands of system and data files, or wiping out hard drives, it is often best to first rebuild the host, then secure the host so that it is no longer vulnerable to the malware threat. Organizations should carefully consider possible worst-case scenarios, such as a new malware threat that necessitates rebuilding a large percentage of the organization's workstations, and determine how the hosts would be recovered in these cases. This should include identifying who would perform the recovery tasks, estimating how many hours of labor would be needed and how much calendar time would elapse, and determining how the recovery efforts should be prioritized.

Determining when to remove temporary containment measures, such as suspended services (e.g., email) or connectivity (e.g., Internet access, VPN for telecommuters), is often a difficult decision during major malware incidents. For example, suppose that email has been shut down to stop the spread of a malware infection while vulnerable hosts are patched and infected hosts undergo individual malware containment, eradication, and recovery measures. It might take days or weeks for all vulnerable hosts to be located and patched and for all infected hosts to be cleaned, but email cannot remain suspended for that period of time. When email service is restored, it is almost certain that an infected host will begin spreading the malware again at some time. However, if nearly all hosts have been patched and cleaned, the impact of a new malware infection should be minimal. Incident response teams should strive to keep containment measures in place until the estimated number of infected hosts and hosts vulnerable to infection is sufficiently low that subsequent incidents should be of little consequence.

Post-Incident Activity



Incident Response Life Cycle (Post-Incident Activity)

Lessons Learned

When a major malware incident occurs, the primary individuals performing the response usually work intensively for days or weeks. As the major handling efforts end, the key people are usually mentally and physically fatigued, and are behind in performing other tasks that were pending during the incident handling period. Consequently, the lessons learned phase of incident response might be significantly delayed or skipped altogether for major malware incidents. However, because major malware incidents can be extremely expensive to handle, it is particularly important for organizations to conduct robust lessons learned activities for major malware incidents. Although it is reasonable to give handlers and other key people a

few days to catch up on other tasks, review meetings and other efforts should occur expeditiously, while the incident is still fresh in everyone's minds. This meeting provides a chance to achieve closure with respect to an incident by reviewing what occurred, what was done to intervene, and how well intervention worked. The meeting should be held within several days of the end of the incident. Questions to be answered in the meeting may include:

- Exactly what happened, and at what times?
- How well did staff and management perform in dealing with the incident? Were the documented procedures followed? Were they adequate?
- What information was needed sooner?
- Were any steps or actions taken that might have inhibited the recovery?
- What would the staff and management do differently the next time a similar incident occurs?
- How could information sharing with other organizations have been improved?
- What corrective actions can prevent similar incidents in the future?
- What precursors or indicators should be watched for in the future to detect similar incidents?
- What additional tools or resources are needed to detect, analyze, and mitigate future incidents?

The lessons learned process for malware incidents is no different than for any other type of incident. Examples of possible outcomes of lessons learned activities for malware incidents are as follows:

- Security Policy Changes

- Awareness Program Changes
- Software Reconfiguration
- Malware Detection Software Deployment
- Malware Detection Software Reconfiguration.

Using Collected Incident Data

Lessons learned activities should produce a set of objective and subjective data regarding each incident. Over time, the collected incident data should be useful in several capacities. The data, particularly the total hours of involvement and the cost, may be used to justify additional funding of the incident response team. A study of incident characteristics may indicate systemic security weaknesses and threats, as well as changes in incident trends. This data can be put back into the risk assessment process, ultimately leading to the selection and implementation of additional controls. Another good use of the data is measuring the success of the incident response team. If incident data is collected and stored properly, it should provide several measures of the success (or at least the activities) of the incident response team. Incident data can also be collected to determine if a change to incident response capabilities causes a corresponding change in the team's performance (e.g., improvements in efficiency, reductions in costs).

Organizations should focus on collecting data that is actionable, rather than collecting data simply because it is available. For example, counting the number of precursor port scans that occur each week and producing a chart at the end of the year that shows port scans increased by eight percent is not very helpful and may be quite time-consuming. Absolute numbers are not informative—understanding how they represent threats to the business processes of the organization is what matters. Organizations should decide what incident data to collect based on reporting requirements and on the expected return on investment from the data (e.g., identifying a new threat and mitigating the related vulnerabilities before they can be exploited.)

Evidence Retention

Organizations should establish policy for how long evidence from an incident should be retained. Most organizations choose to retain all evidence for months or years after the incident ends. The following factors should be considered during the policy creation:

- **Prosecution.** If it is possible that the attacker will be prosecuted, evidence may need to be retained until all legal actions have been completed. In some cases, this may take several years. Furthermore, evidence that seems insignificant now may become more important in the future. For example, if an attacker is able to use knowledge gathered in one attack to perform a more severe attack later, evidence from the first attack may be key to explaining how the second attack was accomplished.

- **Data Retention.** Most organizations have data retention policies that state how long certain types of data may be kept. For example, an organization may state that email messages should be retained for only 180 days. If a disk image contains thousands of emails, the organization may not want the image to be kept for more than 180 days unless it is absolutely necessary. General Records Schedule (GRS) specifies that incident handling records should be kept for three years.

Check Your Understanding (Sorting Activity)

Sort each incident response activity below into the correct phase of the Incident Response Life Cycle.

Preparation

Building Malware-Related Skills

Acquiring Tools and Resources

Facilitating Communication and Coordination

Detection & Analysis

Identifying Incident Characteristics

Identifying Infected Hosts

Prioritizing Incident Response

Containment, Eradication & Recovery

Removal of Malware from Infected Hosts

Rebuilding Affected Hosts

Restoring Full Functionality to the Network

Post-Incident Activity

Lessons Learned

Study Collected Incident Data

Evidence Retention

Quiz

You may take this assessment as many times as needed to acquire a score of 70% or above. The highest score will be the grade recorded in the grade center for this quiz.
Note: You are required to pass each Quiz with an 70% or better in order to gain access to the next chapter/unit within the module.

Question

01/10

According the to NIST definition of an event as “any observable occurrence in a system or network.”

- ◯ True
- ◯ False

Question

02/10

The U.S. National Institute of Standards and Technology defines the incident response life cycle as having four main processes: 1) preparation; 2) detection and analysis; 3) containment, eradication, and recovery; and 4) _____.

incident report

triage

post-incident activity

resolution

Question

03/10

_____ incident responses enables the organization to react to a detected incident quickly and effectively, without confusion or wasted time and effort.

Predefining

Recording

Publishing

Discussing

Question

04/10

_____ is a common indicator of a DoS (Denial of Service) attack.

Unusually light network traffic

Detection of a new virus

User reports of system unavailability

Unknown processes running

Question

05/10

The number-one incident preparation and prevention strategy is _____.

periodic audits of logs

organizational policy

minimize file sharing

configuring network devices

Question

06/10

A(n) _____ is a sign that an incident may have occurred or may be occurring now.

precursor

inactive system

indicator

signal

Question

07/10

A(n) _____ is a sign that an activity now occurring may signal an incident that could occur in the future.

precursor

inactive system

indicator

signal

Question

08/10

If the source of the incident cannot easily be confirmed, it is often better to respond as if it were caused by _____.

an employee

malware

a firewall

a mistake

Question

09/10

The two main aspects of _____ from malware incidents are restoring the functionality and data of infected hosts and removing temporary containment measures.

analysis

detection

recovery

prevention

Question

10/10

Many incident response teams have a prepacked field kit, also known as a(n) _____.

protocal set

evidence kit

forensic bag

jump kit