Interested in learning more about security?

SANS Institute InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Incident Handler's Handbook

One of the greatest challenges facing today's IT professionals is planning and preparing for the unexpected, especially in response to a security incident. An incident is described as any violation of policy, law, or unacceptable act that involves information assets, such as computers, networks, smartphones, etc (Bejtlich, 2005). The scope of this document is limited to the six phases of the incident handling process ("Incident handling step-by-step," 2011) and providing the basic information necessary as to what each ...

Copyright SANS Institute

Author Retains Full Rights

AD

The Incident Handlers Handbook

GIAC (GCIH) Gold Certification

Author: Patrick Kral, patrick.kral@gmail.com

Advisor: Dr. Craig Wright

Accepted: December 5th, 2011

Abstract

One of the greatest challenges facing today's IT professionals is planning and preparing for the unexpected, especially in response to a security incident. An incident is described as any violation of policy, law, or unacceptable act that involves information assets, such as computers, networks, smartphones, etc (Bejtlich, 2005). The scope of this document is limited to the six phases of the incident handling process ("Incident handling step-by-step," 2011) and providing the basic information necessary as to what each step entails. Its overall purpose is to provide the basic foundation for IT professionals and managers to be able to create their own incident response policies, standards, and teams within their organizations. This document will also include an incident handler's checklist (template) that one can use to ensure that each of the incident response steps is being followed during an incident.

© 2012 The SANS Institute Author retains full rights.

1. Introduction

An incident is a matter of when, not if, a compromise or violation of an organization's security will happen. The preparation of the Computer Incident Response Team (CIRT) through planning, communication, and practice of the incident response process will provide the necessary experience needed should an incident occur within your organization. Each phase from preparation to lessons learned is extremely beneficial to follow in sequence, as each one builds upon the other. The following phases will provide a basic foundation to be able to perform incident response and allow one to create their own incident response plan.

2. Preparation

This phase as its name implies deals with the preparing a team to be ready to handle an incident at a moment's notice. An incident can range from anything such as a power outage or hardware failure to the most extreme incidents such as a violation of organizational policy by disgruntled employees or being hacked by state sponsored hackers (Bejtlich, 2005). Regardless of the cause of the incident preparation is the most crucial phase compared to all of the others, as it will determine how well your team will be able to respond in the event of a crises.

There are several key elements to have implemented in this phase in order to help mitigate any potential problems that may hinder one's ability to handle an incident. For the sake of brevity, the following should be performed:

a. Policy – a policy provides a written set of principles, rules, or practices within an organization; it is one of the keystone elements that provide guidance as to whether an incident has occurred in an organization. A login banner can be one way to ensure that individuals attempting to log into an organization's network will be aware of what is expected when utilizing an organization's information assets; for example the login banner (dependent upon the local jurisdiction on privacy) can state that all activities will be monitored and any unauthorized users may civil or criminal penalties, etc. Without clear policies, one could leave their organization legally vulnerable to law suits, such as an employee being fired for looking at porn at work when there was no policy against

© 2012 The SANS Institute Author retains full rights.

such behavior within the organization and provided the opportunity for the said individual to be able to file an improper termination lawsuit (Newman, 2007).

b. Response Plan/Strategy – after establishing organizational policies, now it is time to create a plan/strategy to handle incidents. Prioritization of incidents should be based upon organizational impact (Incident Response Process, 2008), for example a single workstation being non-functional can be considered minor, whereas a server being down could be considered a moderate impact (assuming there are backup servers for failover), and data being stolen directly from human resources that contain privileged information as high. The prioritization of the types of incidents based upon organizational impact can help build the case to receive management buy-in, because without management support then it is likely that the CIRT may not be given the resources necessary to properly handle a crisis.

c. Communication – having a communication plan is necessary, due to the fact that it may be necessary to contact specific individuals during an incident. The entire CIRT should know whom to contact, when it is appropriate to contact them, and why. For example, at 12:31pm the online ordering system for a popular e-commerce site during the holiday shopping season went down and no one is able to access the site; it would be prudent to contact the various individuals with specific expertise to get the system back up and running, as well as management to keep them updated as the situation progresses to resolution. By not having a communications plan, then it is likely that response time will be delayed and/or the wrong people would be contacted and one would not have the proper resources necessary to mitigate the problem (Creating a computer security incident response team: a process for getting started, 2006).

   Please note: it is also necessary to define when it is or is not appropriate to include law enforcement during an incident, due to the consequences that could either positively or negatively affect your organization.

d. Documentation – it is extremely beneficial to stress that this element is particularly necessary and can be a substantial life saver when it comes to incident response. The most significant reason to document an incident is that if the incident is considered a criminal act, then it could be used as evidence to bring the suspect(s) to justice. The other reason for documentation that is just as important is for lessons learned. It is vital that

© 2012 The SANS Institute Author retains full rights.

everything that is done by the CIRT team is documented, that means every action taken (e.g. commands typed, systems affected, etc). Documentation should be able to answer the Who, What, When, Where, Why, and How questions should they ever arise; without such information one could leave a sense of uncertainty if something was called into question, like formatting a hard drive and reinstalling the operating system on a production server (Newman, 2007).

Tip: Checklists with a place for notes, dates and times, and other pertinent information can be extremely handy, especially those that follow each step of incident response.

e. Team – the CIRT should be made up of several people that consist of different disciplines to handle the various problems that could arise during or from an incident. Some of these team members can be attorneys, human resources, public relations, various IT staff with specific specializations, etc. It is particularly beneficial to have an open mind about who is to be included in the CIRT, because some individuals and/or professions may not be as obvious as the actual technical staff to handle an incident (Responding to it security incidents, 2011).

f. Access Control – another key element is to ensure that the CIRT can have the appropriate permissions necessary to perform their job. An excellent example is to have a network or systems administrator being able to add/remove permissions to the accounts of the CIRT during an incident to allow them to mitigate the problem and then have those permissions removed when they are no longer needed (Responding to it security incidents, 2011).

g. Tools – a CIRT without tools is like kayaking without a paddle, it is inherently difficult to get anything done. It is highly recommended having any available software and hardware that can be readily utilized during an incident; this can range from anti-malware to laptops with packets sniffers, screw drivers and other tools, as well as incident response checklists and other items that would be useful (Bejtlich, 2005). All of the tools one would need during an incident should be contained within a "jump bag" that can be quickly grabbed by CIRT members during an incident.

h. Training – this is a must, because without it your team could be ill prepared and result in a complete failure of handling an incident properly despite the best of planning. It is recommended to have drills at regular interval to insure that each individual within the CIRT is able or knows how to perform their duties during an incident.

© 2012 The SANS Institute Author retains full rights.

Jump Bag Recommendations:

- Incident Handlers Journal to be used for documenting the who, what, where, why, and how during an incident.
- Contact list of all CIRT members.
- USB Drives.
- A bootable USB drive or Live CD with up-to-date anti-malware and other software tools that can read and/or write to file systems of the computing environment that the incident response is to be performed in. One example is that of the Bart's PE disk for Windows XP (or later) environments.
- A laptop with forensic software (e.g. FTK or EnCase), anti-malware utilities, and internet access (if necessary for researching solutions or downloading tools).
- Computer and network tool kits to add/remove components, wire network cables, etc.
- Hard duplicators with write-block capabilities to create forensically sound copies of hard drive images.
- A bag that properly store all of the aforementioned tools in an organized and protective fashion.

3. Identification

This phase deals with the detection and determination of whether a deviation from normal operations within an organization is an incident, and its scope assuming that the deviation is indeed an incident. This particular step requires one to gather events from various sources such as log files, error messages, and other resources, such intrusion detection systems and firewalls, that may produce evidence as to determine whether an event is an incident. If a particular event is determine to be an incident, and then it should be reported as soon as possible in order to allow the CIRT enough time to collect evidence and prepare for the preceding steps (Bejtlich, 2005). At this stage of an incident CIRT members should be notified and communication should be coordinated between members along with designated command center staff (e.g. management and/or systems administrators). It is recommended that at least two incident handlers be available to handle an incident so that one can be the primary handler who can identify and assess the

© 2012 The SANS Institute Author retains full rights.

incident and the other to help gather evidence. Communication and coordination between members of the CIRT (and management) is critical, especially if the scope of the incident can have a significant impact on business operations. This is also the phase where incident responders should be documenting everything that they are doing, as stated earlier these documents should be able to answer the Who, What, Where, Why, and How questions in case the documentation is to be used to prosecute the perpetrator(s) in court (Newman, 2007). After determining the scope of the event and documenting the evidence, then the CIRT team can move forward with the next phase (Incident Response Process, 2008).

A good example of the identification phase is a user contacting the help desk and reporting that their system is acting strangely or intrusion detection systems report unusual network traffic from certain hosts. It could come from something ominous as usual activity in system logs that have never appeared before a specific date. It is extremely beneficial to keep a truly open mind the number of possibilities that an incident could be identified. Two other examples worth mentioning are a missing USB drive or other storage media, and a user finding a USB drive somewhere that has public access and plugs into their computer kicking off an auto-run script that steals data or infects systems.

4. Containment

The primary purpose of this phase is to limit the damage and prevent any further damage from happening ("Uf it security," 2011). There are several steps to this phase; however, each one is necessary in order to completely mitigate the incident and prevent the destruction of any evidence that may be needed later for prosecution.

The first step is Short-term Containment; basically the focus of this step is to limit the damage as soon as possible. Short-term containment can be as straightforward as isolating a network segment of infected workstations to taking down production servers that were hacked and having all traffic routed to failover servers. Short-term containment is not intended to be a long term solution to the problem; it is only intended to limit the incident before it gets worse (Bejtlich, 2005).

© 2012 The SANS Institute Author retains full rights.

The second step is System Back-Up; it is necessary before wiping and reimaging any system to take a forensic image of the affected system(s) with tools that are well known in the computer forensics community such as Forensic Tool Kit (FTK), EnCase, et al. The reason behind this is that the forensic software will capture the affected system(s) as they were during the incident and thereby preserving evidence in the event that the incident resulted from a criminal act or to be used for observing how the system(s) were compromised during the lessons learned phase (Newman, 2007).

The last step before the next phase is Long-term containment, which is essentially the step where the affected systems can be temporarily fix in order to allow them to continue to be used in production, if necessary, while rebuilding clean systems in the next phase. Basically the primary focus would remove accounts and/or backdoors left by attackers on affected systems, installing security patches on both affected and neighboring systems, and doing other work to limit any further escalation of the incident while allowing normal business operations to continue (Bejtlich, 2005).

A good example of containment is disconnecting affected systems by either disconnect the affected system's network cable or powering down switches and/or routers to entire portions of the network to isolate compromised systems from those that have not been compromised. This in turn will isolate the problem from the rest of the production network and limit the spread of any malware or reduce the risk of further systems being compromised.

5. Eradication

This phase deals with the actual removal and restoration of affected systems. As with each of the prior phases of incident response, continued documentation of all actions taken will be necessary to determine the cost of man hours and other resources as a means of determining the overall impact to the organization. It is also necessary to ensure that proper steps were taken to remove malicious and other illicit content off of the affected systems, and ensuring that they are thoroughly clean (Incident Response Process, 2008). In general that would mean a complete reimaging of a system's hard drive(s) to ensure that any malicious content was removed and prevent reinfection. This phase is also the point where defenses should be improved after

© 2012 The SANS Institute Author retains full rights.

learning what caused the incident and ensure that the system cannot be compromised again (e.g. installing patches to fix vulnerabilities that were exploited by the attacker, etc).

A good example of actions performed during the eradication phase would be using the original disk images that were created prior to a system being deployed into production to restore the system and then installing patches and disabling unused services to harden the system against further attacks (e.g. disk images created with Clonezilla or Symantec Ghost). One would also scan affected systems and/or files with anti-malware software to ensure any malware that is latent is removed (i.e. using an anti-virus program like Kaspersky combined with CCleaner to disinfect systems and scan the Windows registry for keys that may initiate any latent malware).

6. Recovery

The purpose of this phase is to bring affected systems back into the production environment carefully, as to insure that it will not lead another incident. It is essential to test, monitor, and validate the systems that are being put back into production to verify that they are not being reinfected by malware or compromised by some other means. Some of the important decisions to make during this phase are:

- Time and date to restore operations – it is vital to have the system operators/owners make the final decision based upon the advice of the CIRT.
- How to test and verify that the compromised systems are clean and fully functional.
- The duration of monitoring to observe for abnormal behaviors.
- The tools to test, monitor, and validate system behavior.

There are many more beneficial decisions that could be listed; however, the above information should provide a few ideas as what is entailed. The primary goal overall, as stated earlier, is to prevent another incident from happening that was due to the same problems that cause the one that was just resolved (Responding to it security incidents, 2011).

© 2012 The SANS Institute Author retains full rights.

7. Lessons Learned

The most critical phase after all of the others is Lessons Learned. The purpose of this phase is to complete any documentation that was not done during the incident, as well as any additional documentation that may be beneficial in future incidents. The document should also be written in a form of a report to provide a play-by-play review of the entire incident; this report should be able to answer the: Who, What, Where, Why, and How questions that may come up during the lessons learned meeting. The overall goal is to learn from the incidents that occurred within an organization to improve the team's performance and provide reference materials in the event of a similar incident. The documentation can also be used as training materials for new team members or as a benchmark to be used in comparison in future crises (Bejtlich, 2005).

The lessons learned meeting should be performed as soon as possible; a good rule of thumb is within 2 weeks after the incident. The meeting should go through the incident response report with finalization in an executive summary format. It should be kept short, as to not lose the audience's attention and remain professional.

A good example of performing lessons learned is to have a power point that summarizes the following information:

- When was the problem was first detected and by whom.
- The scope of the incident.
- How it was contained and eradicated,
- Worked performed during recovery.
- Areas where the CIRT teams were effective.
- Areas that need improvement.

It should also include time for suggestions and discussion between members of how to improve the overall team. This phase is extremely beneficial to have members share ideas and information in order to improve team effectiveness in future incidents.

© 2012 SANS Institute, Author retains full rights.

© 2012 The SANS Institute Author retains full rights.

8. Incident Handlers Checklist

1. Preparation
    a. Are all members aware of the security policies of the organization?
    b. Do all members of the Computer Incident Response Team know whom to contact?
    c. Do all incident responders have access to journals and access to incident response toolkits to perform the actual incident response process?
    d. Have all members participated in incident response drills to practice the incident response process and to improve overall proficiency on a regularly established basis?
2. Identification
    a. Where did the incident occur?
    b. Who reported or discovered the incident?
    c. How was it discovered?
    d. Are there any other areas that have been compromised by the incident? If so what are they and when were they discovered?
    e. What is the scope of the impact?
    f. What is the business impact?
    g. Have the source(s) of the incident been located? If so, where, when, and what are they?
3. Containment
    a. Short-term containment
        i. Can the problem be isolated?
            1. If so, then proceed to isolate the affected systems.
            2. If not, then work with system owners and/or managers to determine further action necessary to contain the problem.
        ii. Are all affected systems isolated from non-affected systems?
            1. If so, then continue to the next step.
            2. If not, then continue to isolate affected systems until short-term containment has been accomplished to prevent the incident from escalating any further.
    b. System-backup

© 2012 The SANS Institute Author retains full rights.

i. Have forensic copies of affected systems been created for further analysis?
ii. Have all commands and other documentation since the incident has occurred been kept up to date so far?
    1. If not, document all actions taken as soon as possible to ensure all evidence are retained for either prosecution and/or lessons learned.
    2. Are the forensic copies stored in a secure location?
        a. If so, then continue onto the next step.
        b. If not, then place the forensic images into a secure location to prevent accidental damage and/or tampering.

c. Long-term containment
    i. If the system can be taken offline, then proceed to the Eradication phase.
    ii. If the system must remain in production proceed with long-term containment by removing all malware and other artifacts from affected systems, and harden the affected systems from further attacks until an ideal circumstance will allow the affected systems to be reimaged.

4. Eradication
    a. If possible can the system be reimaged and then hardened with patches and/or other countermeasures to prevent or reduce the risk of attacks?
        i. If not, then please state why?
    b. Have all malware and other artifacts left behind by the attackers been removed and the affected systems hardened against further attacks?
        i. If not, then please explain why?
5. Recovery
    a. Has the affected system(s) been patched and hardened against the recent attack, as well as possible future ones?
    b. What day and time would be feasible to restore the affected systems back into production?
    c. What tools are you going to use to test, monitor, and verify that the systems being restored to productions are not compromised by the same methods that cause the original incident?

© 2012 The SANS Institute Author retains full rights.

d. How long are you planning to monitor the restored systems and what are you going to look for?
e. Are there any prior benchmarks that can be used as a baseline to compare monitoring results of the restored systems against those of the baseline?

6. Lessons Learned
    a. Has all necessary documentation from the incident been written?
        i. If so, then generate the incident response report for the lessons learned meeting.
        ii. If not, then have documentation written as soon as possible before anything is forgotten and left out of the report.
    b. Assuming the incident response report has been completed, does it document and answer the following questions of each phase of the incident response process: (Who? What? Where? Why? And How?)?
    c. Can a lessons learned meeting be scheduled within two weeks after the incident has been resolved?
        i. If not, then please explain why and when is the next convenient time to hold it?
    d. Lessons Learned Meeting
        i. Review the incident response process of the incident that had occurred with all CIRT members.
        ii. Did the meeting discuss any mistake or areas where the response process could have been handled better?
            1. If no such conversations occurred, then please explain why?

9. Windows Template

This template is intended to provide some useful commands and areas to look for anomalous behavior within the Windows environment. Please be aware that these commands are the same as those found on the SANS Intrusion Discovery Cheat Sheet; however this template will take it a step beyond it by providing additional information as to the commands usefulness.

a. To look for unusual processes and services use the following commands:

© 2012 The SANS Institute Author retains full rights.

i. taskmgr.exe – it displays running processes and services.

ii. In command prompt use these three commands:

1. tasklist – it displays a list of running services along with their corresponding PID (process ID), session name, session number, and memory usage. Getting a PID can be useful for using the taskkill command to end the questionable process.
2. wmic process list full – is a windows management interface control that will display all processes, along with detailed information such as their executable path and much more.
3. tasklist /svc – will display a list of all processes along with their corresponding PID, and services that are tied to them.

b. To look for unusual files and registry keys use the Windows search feature and look for files larger than 10MB, and use regedit to look for unusual entries in the following areas:

HKLM\Software\Microsoft\Windows\CurrentVersion\Run

HKLM\Software\Microsoft\Windows\CurrentVersion\Runonce

HKLM\Software\Microsoft\Windows\CurrentVersion\RunonceEx

Generally those three registry entries will contain startup configurations for specific programs, including malware.

c. To look for unusual network usage the following Windows commands in the command line interface (cmd) provides an excellent view of network activity on a particular system:

i. net view \\127.0.0.1 (or localhost) – displays shared folders that are on the system. If there are shared folders that are not supposed to be there that can be a significant red flag.

ii. net session – displays open sessions with other systems on the network. This is useful for detecting communications with other systems on the network and determines whether the connections are legitimate. A good example: is a https connection to a rogue server on the internet and heavy bandwidth usage from the compromised computer in question with that rogue server.

© 2012 The SANS Institute Author retains full rights.

iii. nbstat –S will display NetBIOS activity over TCP/IP on the various network interfaces that a machine in question may have.

iv. netstat and its various flags (e.g. netstat –na, netstat –nao, etc) provides a tremendous amount of information between listening and established TCP/IP connections, along with their ports and whether the protocol used is TCP or UDP. This is useful for determining unusual traffic patterns on the computer in question.

d. To look for unusual start up (or scheduled) tasks, use the following commands:

i. msconfig –displays all startup configurations from services to files in the startup folder, etc. This is also useful for disabling anything trying startup during Windows login or boot-up, and to troubleshoot problems that are caused by nefarious or poorly written programs.

ii. schtasks – displays tasks schedule to run at specific times. This is useful for not only troubleshooting problems, but also looking for would be logic bombs.

iii. wmic startup list full – displays all of the services and programs that startup when Windows boots and/or upon Windows login.

e. To look for unusual accounts use the following three commands:

i. lusrmgr.msc – this command is only useful for looking for local accounts on a machine. Two account types to specifically look for are Administrator accounts that are not supposed to be on the machine and active Guest accounts, as those can lead to serious security compromises.

ii. net user (in command prompt) – displays all user accounts on a local machine.

iii. net localgroup administrators – display all local administrator user accounts. This is useful for finding administrator accounts that do not belong on a particular machine.

f. The final and most crucial area to look for unusual behavior is within event viewer. It displays all of the event log content that Windows actively records. The command for it to type in the run command box is eventvwr.msc.

i. Look for warnings, errors, and other events (e.g. system reboots during usual times, etc).

© 2012 The SANS Institute Author retains full rights.

ii. If the log files are missing, it is a reliable indicator that the machine has been or is compromised and the intruder is trying to hide his\her tracks.

Please consult the "SANS Institute Intrusion Discovery Cheat Sheet for Windows" for more details and updated information at: http://www.sans.org/score/checklists/ID_Windows.pdf

10. UNIX

This template is based off of a similar cheat sheet from the SANS Institute as the one for Windows; however, it is for UNIX based systems. Please be aware that the power behind UNIX systems is in the command line interface (or shell) and therefore it is necessary to be proficient in being able to use it for not only troubleshooting, but also determining whether a system was compromised. Please note that some of these commands may require root (or equivalent) privileges in order to use them (I would personally recommend using the sudo command, if available to perform root required tasks).

a. To look for unusual processes and services use the following commands:
   i. ps -aux – displays running processes along with their process-id (PID), associated user-ids (UID #), name, and other pertinent information. Pay particular attention to any process that is using the UID 0 user-id, because those processes are running with root permissions.
   ii. ps –ef – displays the full listing of all processes and can be useful for finding undesirable processes that are running.
   iii. lsof –p (PID) – displays a specific process in more detail, by displaying the files and ports associated with that process. This is appropriate for examining any Trojan, worms, and other network based malware on a UNIX system.
   iv. lsof +L1 – displays processes running from or accessing files that have been unlinked; basically it will show one to figure out if the attacker is hiding data or running a backdoor.
b. To look for unusual files the "find" command along with its various flags allows one to search a UNIX system for malware. A few examples are listed below:
   i. find / -uid 0 -perm -4000 –print – searches for files that have root permissions.

© 2012 The SANS Institute Author retains full rights.

ii. find / -size +50000k –print – searches for files of a specified or greater size. This is particularly useful for searching for files that may not belong on the system, like movies, games, et al.

c. To look for unusual network usage coming from a system's network interface type in the following command: ip link | grep PROMISC – this command will display any network interfaces that are running in promiscuous mode, which can be a clear indication of an attacker running a packet sniffer.

d. Other useful commands to observe unusual network behavior are:

i. netstat –nap – this displays listening ports and in turn can be useful for finding backdoors.

ii. arp –a – displays all MAC to IP address mappings of the system and can be useful for finding addresses of systems that are not part of the network (e.g. a rouge wireless access point that allows one to gain access into the internal network from the outside).

e. To look for scheduled jobs (i.e. tasks) by root or any other user, type in the following command: contrab –u root –l – this is useful for detecting logic bombs, scheduled connections to unknown hosts, and other potentially nefarious issues.

i. Two additional commands to display system-wide cron jobs are:

1. cat /etc/crontab – displays all jobs scheduled within the cron table.
2. ls etc/cron.* - lists files within the cron subdirectories.

f. To look for unusual accounts use the following commands to check the following files:

i. sort –nk3 –t: /etc/paswd | less – displays all accounts sorted by UID (e.g. UID0, etc), this is useful for finding accounts with root permissions or accounts that do not belong on the system.

ii. egrep ':0:' /etc/passwd – displays only accounts with root permissions.

iii. getent passwd | grep ':0:' – same as above, except for systems with multiple authentication mechanisms.

iv. find / -nouser –print – searches the entire system for orphaned files that may have been deleted by an attacker's temporary account.

© 2012 The SANS Institute Author retains full rights.

g. The best place to check for unusual system activity is the log files, especially in UNIX. Most log files are located in /var/log (or var/logs), or var/messages. A good command to use for viewing log files is: more –f /var/log/messages – this allows a page by page review of all logged events. Pay particular attention to user authentication logins and any unusual patterns such as missing entries and times that may indicate an intruder is trying to hide his/her tracks.

h. Other commands to check for possible clues are:

   i. uptime – displays how long a system has been up and running. If the system's uptime is shorter or longer than it should be, then it could be a clear indication that something has changed and therefore may need further review.

   ii. free – is useful for checking how much ram is used. This is useful for detecting processes that are using a lot of memory (e.g. an attacker searching or modifying a database, etc).

   iii. df – is useful for checking available disk space. This can provide a reliable indication as to whether an attacker is installing malware or removing files from a system.

Please consult the "SANS Institute Intrusion Discovery Cheat Sheet for Linux" for more details and updated information at: http://www.sans.org/score/checklists/ID_Linux.pdf

11. References

Incident response process best practice. (2008, September 25). Retrieved from http://security.tennessee.edu/pdfs/IRPBP.pdf

Responding to it security incidents. (2011). Retrieved from http://technet.microsoft.com/en-us/library/cc700825.aspx

Creating a computer security incident response team: a process for getting started. (2006, February 27). Retrieved from http://www.cert.org/csirts/Creating-A-CSIRT.html

Bejtlich, R. (2005). *The tao of network security monitoring: beyond intrusion detection*. Boston, MA: Pearson Education, Inc.

© 2012 The SANS Institute Author retains full rights.

Incident handling step-by-step and computer crime investigation. (2011). Retrieved from http://www.sans.org/security-training/incident-handling-step-by-step-computer-crime-investigation-day-1-799-cid

Uf it security incident response procedures, standards, and guidelines. (2011, July 13). Retrieved from http://www.it.ufl.edu/policies/security/uf-it-sec-incident-response.html

Newman, R. (2007). *Computer forensics: evidence collection and management*. Boca Raton, FL: Taylor & Francis Group, LLC.

© 2012 SANS Institute, Author retains full rights.

© 2012 The SANS Institute Author retains full rights.

Last Updated: July 23rd, 2016

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

Industrial Control Systems Security Training	**Houston, TXUS**	**Jul 25, 2016 - Jul 30, 2016**	**Live Event**
SANS Vienna	**Vienna, AT**	**Aug 01, 2016 - Aug 06, 2016**	**Live Event**
SANS Boston 2016	**Boston, MAUS**	**Aug 01, 2016 - Aug 06, 2016**	**Live Event**
Security Awareness Summit & Training	**San Francisco, CAUS**	**Aug 01, 2016 - Aug 10, 2016**	**Live Event**
DEV531: Defending Mobile Apps	**San Francisco, CAUS**	**Aug 08, 2016 - Aug 09, 2016**	**Live Event**
SANS Portland 2016	**Portland, ORUS**	**Aug 08, 2016 - Aug 13, 2016**	**Live Event**
SANS Dallas 2016	**Dallas, TXUS**	**Aug 08, 2016 - Aug 13, 2016**	**Live Event**
DEV534: Secure DevOps	**San Francisco, CAUS**	**Aug 10, 2016 - Aug 11, 2016**	**Live Event**
Data Breach Summit	**Chicago, ILUS**	**Aug 18, 2016 - Aug 18, 2016**	**Live Event**
SANS Alaska 2016	**Anchorage, AKUS**	**Aug 22, 2016 - Aug 27, 2016**	**Live Event**
SANS Virginia Beach 2016	**Virginia Beach, VAUS**	**Aug 22, 2016 - Sep 02, 2016**	**Live Event**
SANS Bangalore 2016	**Bangalore, IN**	**Aug 22, 2016 - Sep 03, 2016**	**Live Event**
SANS Chicago 2016	**Chicago, ILUS**	**Aug 22, 2016 - Aug 27, 2016**	**Live Event**
SANS Adelaide 2016	**Adelaide, AU**	**Sep 05, 2016 - Sep 10, 2016**	**Live Event**
SANS Brussels Autumn 2016	**Brussels, BE**	**Sep 05, 2016 - Sep 10, 2016**	**Live Event**
SANS Northern Virginia - Crystal City 2016	**Crystal City, VAUS**	**Sep 06, 2016 - Sep 11, 2016**	**Live Event**
SANS Network Security 2016	**Las Vegas, NVUS**	**Sep 10, 2016 - Sep 19, 2016**	**Live Event**
SANS London Autumn	**London, GB**	**Sep 19, 2016 - Sep 24, 2016**	**Live Event**
SANS ICS London 2016	**London, GB**	**Sep 19, 2016 - Sep 25, 2016**	**Live Event**
Security Leadership Summit	**Dallas, TXUS**	**Sep 27, 2016 - Oct 04, 2016**	**Live Event**
SANS DFIR Prague 2016	**Prague, CZ**	**Oct 03, 2016 - Oct 15, 2016**	**Live Event**
SANS Oslo 2016	**Oslo, NO**	**Oct 03, 2016 - Oct 08, 2016**	**Live Event**
SANS Seattle 2016	**Seattle, WAUS**	**Oct 03, 2016 - Oct 08, 2016**	**Live Event**
SANS Baltimore 2016	**Baltimore, MDUS**	**Oct 10, 2016 - Oct 15, 2016**	**Live Event**
SANS Tokyo Autumn 2016	**Tokyo, JP**	**Oct 17, 2016 - Oct 29, 2016**	**Live Event**
SANS Tysons Corner 2016	**Tysons Corner, VAUS**	**Oct 22, 2016 - Oct 29, 2016**	**Live Event**
SANS San Jose 2016	**OnlineCAUS**	**Jul 25, 2016 - Jul 30, 2016**	**Live Event**
SANS OnDemand	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**