

INCIDENT RESPONSE METHODOLOGY

IRM #6

WEBSITE

DEFACEMENT

Live reaction on a compromised web server

IRM Author: [CERT SG](#)

Contributor: [CERT aDvens](#)

IRM version: 2.0

E-Mail: cert.sg@socgen.com

Web: <https://cert.societegenerale.com>

Twitter: @CertSG

ABSTRACT

This Incident Response Methodology is a cheat sheet dedicated to handlers investigating on a precise security issue.

WHO SHOULD USE IRM SHEETS?

- Administrators
- Security Operation Center
- CISOs and deputies
- CERTs (Computer Emergency Response Team)

Remember: If you face an incident, follow IRM, take notes. Keep calm and contact your business line's Incident Response team or CERT immediately if needed.

→ IRM CERT SG: <https://github.com/certsocietegenerale/IRM>

→ IRM CERT aDvens (French version): <https://github.com/cert-advens/IRM>

INCIDENT HANDLING STEPS

6 STEPS ARE DEFINED TO HANDLE SECURITY INCIDENTS

1. Preparation: get ready to handle the incident
2. Identification: detect the incident
3. Containment: limit the impact of the incident
4. Remediation: remove the threat
5. Recovery: recover to a normal stage
6. Lessons learned: draw up and improve the process

IRM provides detailed information for each step of the incident response process. The steps come from NIST Computer Security Incident Handling Guide.

PREPARATION

OBJECTIVE: ESTABLISH CONTACTS, DEFINE PROCEDURES, GATHER INFORMATION TO SAVE TIME DURING AN INCIDENT.

- Have up-to-date schemes describing your applicative components related to the web server.
- Make sure you have an up-to-date network map.
- Build a backup website up-and-ready, on which you can publish content.
- Define a procedure to redirect every visitor to this backup website (a static maintenance page for example).
- Deploy monitoring and intrusion prevention tools (WAF, fail2ban and the likes) to detect and prevent any abnormal activities targeting your critical webservers.
- Export the web server's log files to an external server. Make sure clocks are synchronized between each server.
- Deploy attack and vulnerability exploitation detection rules based on the server's logs and monitor them.
- Audit your websites before the release and on regular basis (monthly if possible).
- Reference all sources of external static or dynamic contents.
- Have operational contacts of your hosting provider readily available.
- Make sure your hosting provider enforces policies to log all events and verify your contractual compliance.
- Prepare communication templates in case the incident is visible for users and needs to be explained.

IDENTIFICATION

OBJECTIVE: DETECT THE INCIDENT, DETERMINE ITS SCOPE, AND INVOLVE THE APPROPRIATE PARTIES.

Usual channels of detection are:

- Webpage monitoring: The content of a web page has been altered. The new content is either very discreet (an “iframe” injection for example) or explicit (“You have been hacked by xxx”).
- Users: you receive calls from users or notifications from employees about problems they notice while browsing the website.
- Security checks with tools such as Google SafeBrowsing.

Verify the defacement incident and detect its origin:

- Check files’ metadata (in particular, check modification dates, hash signatures).
- Check mashup content providers.
- Check links present in the source code (src, meta, css, scripts, ...).
- Check log files and alerts generated by the detection rules.
- Scan databases for malicious content.

The source code of the suspicious page must be analyzed carefully to identify and scope up the problem.

Be sure the problem originates from a web server belonging to the company and not from the web content located outside your infrastructure, such as in ad banners from a third party.

CONTAINMENT

OBJECTIVE: MITIGATE THE ATTACK'S EFFECTS ON THE TARGETED ENVIRONMENT.

- Back up all the data stored on the web server for forensic purposes and evidence collection. The best practice here, if applicable, is to create a complete bit-to-bit copy of the hard-disk used by the web server. This may notably be helpful to recover deleted content.
- Check your network architecture map. Verify that the vulnerability exploited by the attacker is not located elsewhere:
 - Check the system on which the web server is running
 - Check other services running on that machine
 - Check incoming and outgoing connections made from the server

If the source of the attack stems from another system, investigate the culprit machine.

- Try to find evidence behind every action perpetrated by the attacker:
- Find out how the attacker got into the system in the first place and fix the root cases:
 - A web component vulnerability allowing write access: fix the vulnerability by applying applicable remediations
 - CMS plugin vulnerabilities are often exploited by attackers and need to be identified and patched;
 - Open public folder: make it private
 - SQL weakness allowing injection: correct the code
 - Mashup components: cut off implicated mashup feeds
 - An administrative modification by physical access: modify the access rights

If required (complex issue on an important web server), deploy a temporary up-to-date web server. The server should offer the same content than that one of the compromised machine or at least display legitimate content such as a static maintenance page. The best is to display temporary static content, containing only HTML code. This prevents another infection in case the attacker is still able to leverage the same vulnerability.

REMEDIATION

OBJECTIVE: TAKE ACTIONS TO REMOVE THE THREAT AND AVOID FUTURE DEFACEMENTS.

- Remove all altered content and replace it with legitimate content, restored from earlier backup.
- Make sure this content is free from vulnerabilities; patch if necessary.

RECOVERY

OBJECTIVE: RESTORE THE SYSTEM TO NORMAL OPERATIONS.

- Change all user passwords if the web server provides user-authentication and you have evidence or reasons to believe the passwords may have been compromised. This may require a user communication campaign.
- If a backup server has been used, restore the primary web server components to the nominal state.
- Monitor logs and alerts closely to detect new attacks.

For more details on authentication and infrastructure recovery, check the Large-scale malware compromise IRM-18

LESSONS LEARNED

OBJECTIVE: DOCUMENT THE INCIDENT'S DETAILS, DISCUSS LESSONS LEARNED, AND ADJUST PLANS AND DEFENSES.

Communication

If the defacement has become public, consider preparing and sending out a dedicated communication message explaining the incident.

Report

A crisis report should be written and made available to all of the involved parties.

The following topics should be detailed:

- Initial detection
- Actions and timelines
- What went right
- What went wrong
- Incident's cost
- Indicators of compromise

Should a vulnerability be identified, report any undocumented flaw impacting to the application's editor, so that the code can be reviewed and receive an official fix.

Capitalize

Actions to improve the handling of defacement incidents should be defined to capitalize on this experience.