Preparation 1

Objective: Establish contacts, define procedures, and gather information to save time during an attack.

- Have up-to-date schemes describing your applicative components related to the web server.
- Build a backup website up and ready, on which you can publish content.
- Define a procedure to redirect every visitor to this backup website.
- Deploy monitoring tools to quickly detect any abnormal behaviour on your critical websites.
- Export the web server's log files to an external server. Make sure clocks are synchronized between each server.
- Reference external contents (static or dynamic) and create a list for each of them. Don't forget third parties for advertisement.
- Reference contact points of your hosting provider.
- Be sure your hosting provider enforces policies to log all events.
- Make sure you have an up-to-date network map.

Identification 2

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 obvious (*"You have been 0wn3d by xxx"*)
- User: users call or notification from employees about problems they noticed while browsing the website.
- Security checks with tools such as Google SafeBrowsing

Verify the defacement and detect its origin:

- Check files with static content (in particular, check the modification dates, hash signature).
- Check mashup content providers.
- Check link presents in the web page (src, meta, css, script, …).
- Check log files.
- Scan the databases for malicious content.

The source code of the suspicious page must be analysed carefully to identify the problem clearly. In particular, **be sure the problem is on a web server belonging to the company** and not on a web content located outside your infrastructure, like commercial banners from a third party.

Containment 3

Objective: Mitigate the attack's effects on the targeted environment.

- **Backup all data** stored on the web server for forensic purposes and evidence collecting. The best practice here if applicable is to make a complete bit-by-bit copy of the hard-disk containing the web server. This will be helpful to recover deleted files.
- **Check your network architecture map. Verify that the vulnerability exploited by the attacker is not located somewhere else :**
  - Check the system on which the web server is running,
  - Check other services running on that machine,
  - Check the connections to other systems, which might be compromised.

If the source of the attack is another system on the network, disconnect it if possible physically and investigate on it.

Try to find evidences of every action of the attacker:

- **Find out how the attacker got into the system in the first place and fix it :**

- Web component vulnerability allowing write access: fix the vulnerability by applying editor's fix.
- Open public folder: fix the bug.
- SQL weakness allowing injection: correct the code.
- Mashup components: cut mashup feed.
- Administrative modification by physical access: modify the access rights.

- **If required (complex issue and very important web server), deploy a temporary web server**, up to date with its applications. It should offer the same content than the compromised web server or at least show another legitimate content such as "Temporary unavailable". The best is to display a temporary static content, containing only HTML code. This prevents another infection in case the attacker has used vulnerability in the legitimate PHP/ASP/CGI/PL/etc. code.

Remediation 4

Objective: Take actions to remove the threat and avoid future defacements.

Remove all altered content and replace it with the legitimate content, restored from earlier backup. Make sure this content is free from vulnerabilities.

Recovery 5

Objective: Restore the system to normal operations.

- **Change all user passwords**, if the web server provides user-authentication, and you have evidence/reasons to think the passwords may have been compromised. This can require a large user communication
- **If backup server has been used, restore the primary web server component as nominal**

Aftermath 6

Objective: Document the incident's details, discuss lessons learned, and adjust plans and defences.

Communication

If the defacement has been visible for part of your users, plan to explain the incident publicly.

Report

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

The following themes should be described:

- Initial detection;
- Actions and timelines;
- What went right;
- What went wrong;
- Incident cost.

In case of vulnerability discovery, **report any undocumented vulnerability** lying on a product running on the web server (like a PHP forum) to its editor, so that the code can be upgraded in order to release a fix.

SOCIETE GENERALE | CERT SOCIETE GENERALE

Incident Response Methodology

IRM #6

Website Defacement

Live reaction on a compromised web server

IRM Author: CERT SG / Cédric Pernet
IRM version: 1.3

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 and do not panic. Contact your CERT immediately if needed.

Incident handling steps

6 steps are defined to handle security Incidents

- **Preparation: get ready to handle the incident**
- **Identification: detect the incident**
- **Containment: limit the impact of the incident**
- **Remediation: remove the threat**
- **Recovery: recover to a normal stage**
- **Aftermath: draw up and improve the process**

IRM provides detailed information for each step.