1 Introduction

Hi my name is Mark Bellingham and I'm here today to demonstrate my program, which is on the topic of Website Security and will focus on the area of detecting and preventing website defacement

2 Why do we need website security?

Hacked websites can

- **Cause embarrassment**
  - Makes your company look incompetent, which is bad if you handle sensitive data – and who doesn't these days? Can use your resources to promote a political message that you don't agree with.
- **Prevent users from accessing your content.**
  - If you are an online business or rely on your website, this can lead to a loss of revenue while your site is down, and even more if you have to pay someone to put it back up again.
- **Potentially puts your user's security at risk.**
  - Forms could be modified to redirect your user's personal and private information. We have all heard of phishing attacks where people try to persuade your users to visit a fake website but no-one imagines that the official site could be leaking your data.
- **Let people steal your data.**
  - If hackers have access to the website they could have access to other parts of the system too
- **Be used to launch malicious software**
  - This could materialise in many forms, from email spamming, being part of a bot network that launches DDOS attacks, delivering viruses and Trojans to your customers, to name just some of the potential problems

3 Examples of hacked websites

Hacked websites is not just something that amateurs have to deal with. Even the richest companies get hit sometimes. Here are two examples, from Microsoft and Google. It just goes to show that everybody needs to think about website security, from the smallest company site to the biggest multi-national enterprise.

4 Good security principles

The problem of website hacking is something that will always exist. As long as websites are connected to the internet, people will be looking for security holes. It won't be possible to find them all and adding new features only serves to add more potential for vulnerabilities.

There is no one single solution when it comes to security, in all areas not just website security. A single solution trying to cover all bases only makes it more likely to fail in some areas.

The best approach is a layered one, with small pieces of software targeting a particular line of attack and doing it well. That will be the focus of the program in this project. To identify when the website has been illegally modified, and try to put it right.

5 Aims of the program

Download and hash source code

Monitor website for changes

Identify users at the time of change

Notify the website administrator

Restore the website

Try to implement basic blocking of repeat offenders

6 Program design

In designing the program the use cases were identified, the website itself is a simple four-page site with some basic formatting. The flow diagram shows how the program moves from one action to another. So here is the section where it is initialising, creating and storing the first hashes. Next is the section for monitoring. The program remains in this section until it identifies a page modification. If such a state is found, these are the next steps is should take before returning to the monitor.

7 Program features

Mainly text based output.

- This means that it can easily be integrated into another service or program

Key events emailed to the administrator

- gives one central location for monitoring reports as well as a quick notification so that the admin is kept informed.

Lightweight on resources

- should not impact performance of the web server

Minimal input from the user

- It really is just setup and go. No need for the user to do anything after the initial setup

Continuous Monitoring even after an attack

- The program resets itself even after a breach, meaning that if the administrator cannot reach the server at that time, they can rest assured that the website is continuing to function as intended.

Permanent store of IP addresses and timestamps

- which the administrator can refer to if they find that they need to improve security in other areas

8 How to use the program

Input list of pages into the database

Start running once your website is complete

Stop the program then restart after intentionally changing your website

9 Page monitoring

So the program will connect to the website, download the code for each page and create a hash of it. A hash is a one-way fixed-length encryption format that always generates the same result when given exactly the same source, but generates very different output with even just a slight change in the source. The hashes will be stored and used as the reference point for the entire time that the program is running, until it is restarted whereupon the hashes are regenerated.

The program then continues to download the code for each page and hashing it at intervals of one minute. These hashes are compared with the original set that were generated the first time round. If there is no change the program repeats this section ad infinitum.

If there is a change in the hash this triggers the next part of the code to run. The program generates the current time, gets the access_log from the web server, finds any entries that have the same or one minute previous, and gets the IP address that is associated with that entry. These IP addresses are stored in the database so that they can be referred back to later, even after a program restart.

10 Notifying the administrator

The program bundles up some useful information for the administrator which includes the name of the page that has been changed, the two hashes which prove that there has been a change, the time of the modification, the IP addresses that were accessing at that time, and whether those IP addresses are known to the system or not.

The program then restores the website to how it was originally, by copying over the files from the backup location and continues the monitoring process.

11 Identifying a hacker’s IP address

If the program finds another modification it goes through the previous processes of trying to identify the IP address, notifying the administrator and restoring the website. The difference here however is that if the found IP is one that is already known to the system, the program informs the admin and adds that address to the .htaccess file that is associated with the user.

12 Blocking the hacker

Here is an example of what the user will see if they have been blocked from accessing the website.

13 What the program does well

Identifying when web pages have been modified

Identifying who was using the website at that time

Alerting the administrator

Restoring the website

14 What it can’t do

Block users who access via SSH etc

In order to block users who are accessing the website files though something like Secure Shell, the administrator would need to set up rules in the IP tables of their firewall or network monitoring program. This is beyond the scope of what this program is trying to achieve. This program only ever tries to create a simple block with the aim of deterring the type of hacker who will get bored of seeing their efforts undone and then not being able to see anything at all. For a more determined hacker the program provides some useful information that will help the administrator deal with that threat.

Cope with DDOS attack

If an attacker is using a bot net to attack the website continuously, this program will struggle to defend against this type of attack. Again defending against DDOS is really the realm of much more specialised software with the help of the firewall. As stated earlier, good security is made up of multiple layers that come together to create something that is more than the sum of their parts.

Identify an attacker who is using the host machine

If the system did this then it would immediately block itself from working. If an attacker has access to the host machine then the security problems are much more serious than a defaced website.

15 Demonstration

Now I am ready to perform a demonstration of the program. Does everyone understand the purpose and function of the program from the explanation so far?

16 Any questions?

At this point I should make clear that the program is designed and intended to run on a single machine. There are far too many variables for things like fixed file locations and making a program that will run on any machine would be much more complicated – needlessly so. Submitted with the program are instructions for getting it working on your machine but it would be much easier for me to answer any questions or demonstrate any further features while we have it running here in the Lab.