

# Detecting and Containing IRC-Controlled Trojans: When Firewalls, AV, and IDS Are Not Enough

Corey Merchant 2002-07-10

## Detecting and Containing IRC-Controlled Trojans: When Firewalls, AV, and IDS Are Not Enough

by Corey Merchant and Joe Stewart, LURHQ Corporation Secure Operations Center

last updated July 10, 2002

---

### Abstract

This paper discusses IRC-based trojans as a distinctly underestimated class of malicious activity, and how real time security event monitoring is the key to identifying and containing similar compromises. It discusses the general methodology used to discover, track, and stop such malicious activity by presenting a real-world case study.

### The Incident

The firewall alerts, as shown in Table 1 began around midnight: prolific scans from a host inside one of our customers' perimeter to many Internet hosts for IRC (TCP 6667). Within minutes we had thousands of alerts. And in a few days we were well into exploring our attacker's methods, tools, and activities - an endeavor that would result in a greater awareness of what I now appreciate as a distinct class of attacks (IRC based trojans), and re-enforcement of the fact that firewalls, anti-virus software, and intrusion detection systems are not enough. Constant security event monitoring is the missing key to most information security infrastructures.

Table 1. Firewall Logs

```
Apr 25 18:15:14 customer_firewall unix: securityalert: tcp if=hme1 from xxx.xxx.
xxx.43:3622 to
212.110.161.45 on unserved port 6667
Apr 25 18:04:32 customer_firewall unix: securityalert: tcp if=hme1 from xxx.xxx.
xxx.43:3690 to
203.121.68.219 on unserved port 6667
Apr 25 17:55:15 customer_firewall unix: securityalert: tcp if=hme1 from xxx.xxx.
xxx.43:4240 to
209.116.7.23 on unserved port 6667
Apr 25 17:56:48 customer_firewall unix: securityalert: tcp if=hme1 from xxx.xxx.
xxx.43:4802 to
212.74.101.21 on unserved port 6667
```

..and so on over thousands of hosts.

### The Attacker's Approach

This attacker's code is probably unique, but the approach is not new or unique. Still, you seldom hear of it -

the use of semi-custom trojan packages using IRC as a control channel. Some alerting organizations allude only to increases in IRC-based backdoors and malware, but do not mention the individual trojan package(s) used. And most virus protection packages catch very few tools like this. The reasons for this is that there are too many such tools, most are evolving, and their propagation is more deliberate and controlled, unlike newsworthy worms like Nimda and Melissa. There are likely hundreds, perhaps thousands, of these types of packages, which may really be considered crackers' tool kits - Swiss Army knives for cracking, if you will. They range from the collected and self-written scripts of individuals to collections of such tools available for download and use by less skilled attackers. Most work like this:

- Infect a host (usually one with good bandwidth and high level of anonymity, like a DSL or cable modem home user). This may be done in a variety of ways, usually cracker 101 level activity.
- Add a variety of tools for common tasks (backdoors, DOS, host and port scanning, hiding the attacks identity, scuttling the host if necessary, etc.).
- Load some control code (often a collection of mIRC scripts).

This control code is what really defines this class of trojan package. The control code provides the attacker a means to run and control various utilities via IRC, backdoor access to the host, and some mechanism for file transfer. More sophisticated packages allow their victims, or zombies, to report back regarding the success/progress of attacks. This allows the attacker to control dozens and, in some cases, hundreds or thousands, of zombies with a great deal of flexibility and vision. A great majority are centered around a hacked version of mIRC, a common IRC client; and associated mIRC scripts. To summarize, the power of this approach is three-pronged. First, it uses commonly available tools to completely control victims. Second, it uses an array of intermediate victims to obfuscate the attacker's identity. And third, there are so many similar tool kits out there that they provide an overload of small moving targets for the good guys (virus protection vendors, firewall vendors, and security staffs in general). As an example, neither CERT nor the FBI were able to appoint any resources to this incident.

## Discovering the Compromise

After the first wave of scans, we asked that the host be checked out thoroughly. It turned out to be the laptop of a remote worker with DSL who had remote access through a VPN (split tunneling disabled). The machine had current anti-virus software and a personal firewall. The user was described as non-technical, which led us to believe that the laptop had been compromised in one way or another, so it was sent in for a scrub down. Nothing was found. The group that did this said that the box was re-imaged, but when it came back up on the network we immediately began getting scans from it to many Internet addresses for Squid (TCP 3158) and SQL (TCP 1433). We asked that the box be sent to us this time.

We set the machine up on an outside network alongside a sniffer, booted up, and captured for an hour. At this time we notice that it was SYN flooding a Web server (see Table 2). Upon further investigation using Ethereal and other tools, we discovered that it and about 75 other zombies were taking part in a distributed denial of service attack, all controlled from an IRC channel on a server presumably in Kazakhstan. Therefore, we had to take the host off-line. But, we had enough captures to see that it was taking part in distributed

scans for certain services (looking for new zombie recruits), was transferring and receiving files from its master (mostly scripts to perform all these tasks and to scuttle the host itself if necessary), as well as the DDOS activity mentioned. From the information gathered, we believe that the activity ties together something like this. The attacker:

- compromises new hosts via a vulnerability in SQL, using it to run remote commands that download and install the trojan package. All the victims we saw were running SQL servers that had SA accounts with no password. All were DSL or cables modem users. This is the port 1433 scans we saw.
- hides their identity by hopping through other hosts, usually open Squid proxies. This is the port 3158 scans we saw. See Table 3 for capture decodes of how our attacker scans for more open Squid relays.
- controls and communicates with the zombies via an IRC channel on a valid, but compromised server.

Table 2. DDOS Capture Decode

```
# Stran is the op, our attacker. Here he sets the channel topic, which tells the
# zombies who to attack, and on what port(s).
:Stran!~stran@fbi.gov TOPIC #rubik :!0pana www.gotocasino.com 80
# A zombie on attbi.com's net reports that it is "bOmBiNg" the victim.
:xJ9XI58!~F107638I@rox-21043.atl.client2.attbi.com PRIVMSG #rubik
:5,1[#4bOmBiNg#5]#0-#3[#09 www.gotocasino.com #3]
# The same zombie reports back a successful attack
:xJ9XI58!~F107638I@rox-21043.atl.client2.attbi.com PRIVMSG #rubik
:14Attacked host #15: #4 www.gotocasino.com #14port #15: #4 80
# This goes on for many hosts, who continue to syn flood the victim.
```

Table 3. Distributed Squid Scan

```
# The channel op Stran, our attacker, tells the zombies to scan the addresses
and
ports in
# in 3128.txt
:Stran!~stran@fbi.gov PRIVMSG #rubik :!0 play #rubik 3128.txt 2500
PRIVMSG Stran :##00X# #U##15nderstood
```

Important notes here are that the network-based IDS missed this activity because there were no signatures for this specific trojan, the virus software on the host also missed it for similar reasons, and the alerts on the firewall looked upon first glance like standard internal network noise. However, the security staff, watching in real time, clued in that the source was a VPN network, and that the destination was a bit suspicious.

## Anatomy of the Trojan

It is useful to understand what some of the standard pieces of one of these trojan are. The following is from [Trend Micro's virus encyclopedia](#). It outlines what they said about the trojan after we had turned over the files. They and other virus vendors have since released pattern updates for this trojan.

*Upon execution, this backdoor malware extracts the following files:*

```
C:\Winnt\system32\ght.dll
C:\Winnt\system32\hajr.drv
C:\Winnt\system32\iexplorer4.cab
C:\Winnt\system32\Msboot.exe
C:\Winnt\system32\msflxgd.exe
C:\Winnt\system32\smtp.txt
C:\Winnt\system32\syscribe.txt
```

*It then executes Msboot.exe, a file written in Visual Basic, which is a loader program. To allow itself to execute upon subsequent reboot, it creates a registry run key as follows:*

```
HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsCurrentVersion\RunServices
?Microsoft ReBoot?= ?C:\Winnt\system32\Msboot.exe?
```

*MSFLXGD.EXE is a modified PE-Compact mIRC client. Upon execution, it drops a configuration script, netv.ocx (detected as IRC\_SOER.A). With it and the other accessory scripts, it functions as a backdoor program connecting to the site Irc.Kamaz.Kz via port 7777. There, it joins a specific channel and sends messages to the hacker's nick. It also waits for specific keyword triggers that serve as commands for it process.*

*When this program is running on the infected computer, the hacker can execute the following:*

*Port scan*

*Launch a TCP and UDP flood*

*Initiate fileserver*

*Initiate an icq page bomb*

*Edit registries*

*Format the infected user's Hard Drive*

*Connect to a URL*

*List the infected user's drives*

*In addition to the abilities that Trend mentions, we have seen the following:*

- The ability to mail bomb domains.
- The ability to scan specifically for SubSeven, including connection-banner checking.

*And, we have seen our attacker add and remove special-function scripts periodically.*

## **The Victims**

*It appears that this attacker's primary means of propagating the trojan was by exploiting open MS-SQL servers running on Windows NT/2000 boxes. All zombies had SQL servers with SA accounts that had no password. We were able to recreate similar activity using the command shell provided by MS-SQL. All zombies were also*

*attached to DSL routers or cable modems, so our attacker appears to be interested in bandwidth, as well. Our methods were:*

- Re-writing the mIRC scripts to disable malicious functionality.
- Logging into the channel, issuing a special command culled from the original scripts that allowed us to have full control over the zombie network.
- Ran the comomand 'info', which displayed the IP addresses of all logged-in zombies.
- Ran a PERL script to get SMTP banners from a sample of zombies (see Table 4).
- Port scan one zombie and found a number of interesting ports, but clued into 1433 since one of the original scans was for this port. It was MS-SQL running. The SA account had a null password. We check a sample of the other zombies and found this to be true for all.
- Normal host identification techniques (nslookup, whois, nmap. etc.)

Table 4. SMTP Banners

```
194.102.126.7 Banner: 220 sediu Microsoft ESMTP MAIL Service, Version:
5.0.2195.4905 ready at Wed, 8 May 2002 14:59:08 +0300
206.132.210.10 Banner: 220 risnt7.lefrak.com ESMTP Server (Microsoft Exchange
Internet Mail Service 5.5.2650.21) ready
24.228.6.199 Banner: 220 ramaposoftware.BEXTInc.com Microsoft ESMTP MAIL
Service, Version: 5.0.2195.2966 ready at Wed, 8 May 2002 07:51:10 -0400
24.90.7.117 Banner: 220 computech-srvr.computechny.com Microsoft ESMTP MAIL
Service, Version: 5.0.2195.4905 ready at Wed, 8 May 2002 07:52:43 -0400
24.98.120.133 Banner: 220 infoe.infoequation.com Microsoft ESMTP MAIL Service,
Version: 5.0.2195.2966 ready at Wed, 8 May 2002 07:53:37 -0400
24.98.20.168 Banner: 220 server20002.DOMAIN2000 Microsoft ESMTP MAIL Service,
Version: 5.0.2195.2966 ready at Wed, 8 May 2002 07:46:34 -0400
61.79.104.99 Banner: 220 minserver1 Microsoft ESMTP MAIL Service, Version:
5.0.2195.2966 ready at Wed, 8 May 2002 20:51:27 +0900
64.210.163.25 Banner: 220 webdemo Microsoft ESMTP MAIL Service, Version:
5.0.2195.4453 ready at Wed, 8 May 2002 04:54:29 -0700
64.210.163.75 Banner: 220 rational2.i-telco.com Microsoft ESMTP MAIL Service,
Version: 5.0.2195.2966 ready at Wed, 8 May 2002 04:55:12 -0700
64.210.163.78 Banner: 220 SLIN.i-telco.com Microsoft ESMTP MAIL Service,
Version: 5.0.2195.1600 ready at Wed, 8 May 2002 04:55:29 -0700
```

*Finally, we e-mailed the most relevant contact available for the victim hosts we had on record, informing them of the activity we had seen and that they had vulnerable and compromised hosts.*

## **Conclusions**

*Anecdotal evidence based on activity we've seen on over 100 networks based solely in the United States, along with information gathered during this incident, suggests that these IRC-based trojans and similar tool kits likely account for a vast majority of deliberate, successful security breaches. Yet, most go unnoticed. Further, these are generally not that sophisticated. Merely taking the technique up one level of expertise - for example, by adding encryption and communication/control over port 443, which almost no one filters or inspects*

*outbound, or encoding and tunneling in another valid protocol - makes the existence of the ongoing penetration nearly invisible. In a corporate or government espionage scenario, one can envision an attacker silently gathering proprietary information over a long period completely unnoticed. In a warez scenario, these kinds of zombie networks are the latest trend in storing, moving, and distributing pirated software and media.*

*Given that the current standard security model for most organizations is product based, this type of activity is routinely overlooked. Most of these organizations bought firewalls and virus software during a big security push in the early 1990s, and intrusion detection systems later, but none of these products detected this trojan and none stopped it, and general awareness in security circles is surprisingly low. Mass media outlets are inclined to cover large scale worm outbreaks, while most security types tend to focus on one dimensional attacks (e.g. someone is port scanning the firewall).*

*As we've seen, there are a few components necessary for discovering and stopping this class of attack:*

- Products - the normal array of firewalls, sniffers, application/OS logging, and intrusion detection systems.
- Reactive abilities - using a sniffer and understanding the output, understand firewall and IDS alerts.
- Reverse engineering the trojan binaries, scripts, and configuration files.

*However, none of these in isolation are effective. And, more importantly, the critical glue that binds these is active, real-time monitoring of security events 24x7. The diligent watching of trained security staff is the most effective real defense for this type of activity.*

*Corey Merchant works for [LURHQ Corporation](#), a managed security services provider where he functions as Secure Operations Center Manager and Senior Analyst.*

[Privacy Statement](#)

Copyright 2006, SecurityFocus