

Remote Shell Trojan: Threat, Origin and the Solution

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/incidents/2001-09/0085.html>

From: kai takashi (rst@coders.com)

Date: 09/09/01

From: kai takashi <rst@coders.com>
To: bugtraq@securityfocus.com
Subject: Remote Shell Trojan: Threat, Origin and the Solution
Date: Sun, 9 Sep 2001 14:40:27 +0300
Message-Id: <01090914541500.08399@bandit>

Overview:

At the 5th of September Qualys released a Security Warning regarding a Linux based virus. This virus was called the "Remote Shell Trojan" (RST) and it attacks Linux ELF binaries. It has replicating abilities: when run it will infect all binaries in /bin and the current working directory. Besides that it also spawns a process listening on UDP port 5503. When a properly crafted packet is received by this process it will connect back with a system shell.

Danger:

Very often viri are not seen as a real security threat for UNIX. A virus can not infect binaries where the userID it is running under has no write access to. Even under this situation viri can be a threat for UNIX based operating-systems: Everytime a infected binary is run it will infect all binaries in the current working directory. It is not unthinkable that a user with increased privileges will later run a binary infected by the RST. In this way the virus can transparently spread itself over the system. This is especially the case in production environments of in an environment where many users share files. This process will get into a rapid once the /bin binaries are infected. Every execution of normal system commands like 'ls' will infect all binaries in the current working directory. In spite of the theoretical immunity UNIX has is the situation described here not unlikely to happen in many human situations. The backdoor process can give unprivileged people access to your system under the UserID the backdoor process is running. Attackers can attempt to get higher privileges on the system from there.

Origin:

RST was developed by us as a research project and intended only for internal use on our systems. Our goal was to analyse how a non-privileged virus could

SecurityFocus Incidents: Remote Shell Trojan: Threat, Origin an

affect a system running Linux in a normal work–environment. Things however didnt go as they were intended to go. An infected binary accidentally leaked out our research lab and came into the hands of so called "scriptkiddies". They infected their own systems and other systems where they had access to. From this point the virus seemed to spread in the wild. This should never have happened and we truly apologize that it did.

Our main concern now is that the spread of this virus gets stopped and that all the infected hosts get cleaned as soon as possible. As of now the format of the specially crafted packet send to the listening backdoor process is unknown to the public. But this might eventually get reverse engineered in the future and RST can then be actively abused by other people.

Solution:

We have created a set of utilities which can recursively detect and remove the virus from the system. It also has the option to make binaries IMMUNE for future infection by the RST. We put our best effort in making these utilities as easy to use as possible. And we **STRONGLY RECOMMEND** that you run these to see if you are infected and to remove the RST from all the infected binaries. We especially recommend that multiuser systems make their system immune for the RST as the risks for these systems are much higher. Immunisation works by increasing the size of the text segment by 4096 bytes so that the "hole" between the text and data segments is gone. After this there's no space for the RST to add it self to the binary anymore.

The interface to these programs is simple and self–explanating. The user can decide wether he wants to automatically detect and remove the RST on the system recursively or if he wants to apply the remover on a per binary base. In this mode he can also get a individual status report on wheter this binary is infected, immune or innocent. Sample usage would be:

```
% perl Recurse.pl remove
```

For more information regarding this read the included documentation.

Conclusion:

Again we strongly recommand that anybody running Linux runs the detector to see if their system is infected. Even if they do not expect anything, they can always optionally immunise their system. This is the only way we can fight the further spread of this virus. Again we apologise for all the inconvenience this may have caused. But maybe we can see it as a lesson that Linux and UNIX are not immune for viri.

Regards,
– anonymous

SecurityFocus Incidents: Remote Shell Trojan: Threat, Origin an

- application/x-gzip attachment: Kill the beast!

This list is provided by the SecurityFocus ARIS analyzer service.
For more information on this free incident handling, management
and tracking system please see: <http://aris.securityfocus.com>

- **Previous message:** Kent Engström: "Re: Pretty stealthy SSH scanning seen on the Internet."
- **Next in thread:** Nick FitzGerald: "Re: Remote Shell Trojan: Threat, Origin and the Solution"
- **Reply:** Nick FitzGerald: "Re: Remote Shell Trojan: Threat, Origin and the Solution"
- **Reply:** John Stauffacher: "RE: Remote Shell Trojan: Threat, Origin and the Solution"
- **Reply:** Patrick Andry: "Re: Remote Shell Trojan: Threat, Origin and the Solution"
- **Reply:** Kevin Gagel: "Re: Remote Shell Trojan: Threat, Origin and the Solution"
- **Messages sorted by:** [date] [thread] [subject] [author] [attachment]