CUC - 一种[1]自扩散的网络攻击系统

芮苏英 龚俭

（东南大学 计算机系，210096 南京）

【摘要】本文详细分析一种具有自扩散能力的攻击系统 CUC 的行为模式和体系结构，并且研究了它的扩散机制。此外，对该系统所利用的漏洞作了完整的分析，剖析了该系统与 1988 年“蠕虫”相比的特点。

【关键字】网络安全 扩散性 蠕虫

CUC- A Self-pervading Network Attacking System

Rui Suying GongJian

(Southeast University, Computer Science Dept. , 210096 Nanjing, P.R.China)

【Abstract】The behavior pattern and the architecture of a self-pervading network attacking system -CUC is analyzed in this paper, and its pervading mechanism is also studied. Besides, this paper provides a complete description of the vulnerabilities exploited in this system and comparison between CUC system and Worm which was spasmed in 1988.

【Key words】 Network Security Pervasion Worm

1. 引言

近期，NJCERT 收到多起的网络安全事故报告称：有系统对外部主机的 111 端口和 80 端口进行大规模的横向扫描，扫描动作不带隐蔽性、扫描持续时间较长。NJCERT 技术人员为发起扫描的此类主机进行了系统的安全检查，发现此类主机实际是已被攻破的受害者，攻击者在这些主机中植入攻击代码。这些被植入的攻击代码启动了扫描进程，对外进行大规模的扫描。

通过对植入系统的攻击代码进行分析，发现这些代码具有自扩散性：当攻击者攻破第一台受害主机，攻击代码被注入系统；这些攻击代码在合适的时机利用一级受害主机的系统资源，启动对其他的主机的攻击，并且对二级受害主机同样注入攻击代码。利用这样的手段，攻击在 Internet 中不断扩散，造成比较大的影响。

攻击代码被激活后以独立进程的方式潜伏在系统中，并且感染其他系统。这种运行和传播方式与“蠕虫”的工作原理相同。从功能上看，它已经成为一个网络攻击的系统，具有一定的结构和特定的工作方式。

该网络攻击系统一般将攻击代码放置在受害主机的/dev/cuc 目录下，因此被暂时命名为 CUC 网络攻击系统(也有其他变种版本自称为 Chinaworm)。

本文对该攻击系统作比较详细的分析，由于众所周知的原因，攻击代码细节部分从略。

2. CUC 网络攻击的分析

2.1 CUC 网络攻击的行为描述

首先，对 CUC 网络攻击系统作宏观上的分析。

假设一级受害主机的入侵已经完成，我们从它的扩散入手，观察该系统对整个网络产生影响步骤（为描述方便，将被扩散主机成为二级受害主机）：

第一步，随机生成 IP 作为二级受害主机的超集，对这些 IP 所在的 C 类网段的 111 口进行横向扫描（检测是否存在 rpc 服务），保存结果，获得存在 rpc 服务的主机集合；

第二步，对上述存在 rpc 服务的主机，检测是否运行 sadmind 服务，保存检测结果，获得存在 sadmind 服务的主机集合，即二级受害主机集合一；

第三步，对二级受害主机集一以轮询方式尝试 sadmind 栈溢出攻击；

第四步，检查栈溢出攻击是否成功。如果成功则进行第五步（扩散攻击系统）否则跳第九步（进行另一种攻击尝试）；

---

[1] [1]作者简介： 芮苏英，硕士研究生，主要研究方向为网络安全。 龚俭，工学博士，东南大学计算机系教授、博士生导师，主要研究方向包括网络管理、网络安全、网络体系结构、开放分布式处理等。

第五步，sadmind 栈溢出攻击成功后，攻击程序可以通过登录二级受害主机的 600 口获得一个具有超级用户权限的 shell。攻击程序利用这个 shell，添加二级受害主机对一级受害主机的信任关系，使一级受害主机可以执行二级受害主机的远程 shell 指令。

第六步，一级受害主机将攻击代码上载至二级受害主机，设置二级受害主机的攻击环境，修改系统启动文件并且消除入侵痕迹。

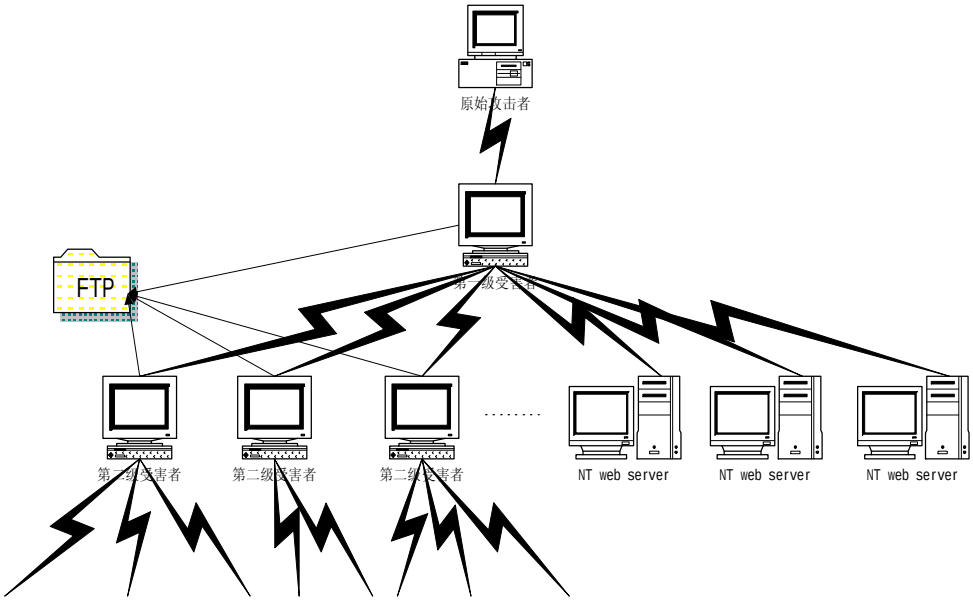
第七步，一级受害主机远程启动二级受害主机的攻击进程。由于在第六步中，一级受害主机设置了二级受害主机的攻击环境、修改了系统配置，因此，二级受害主机不能通过重新启动系统阻止 CUC 攻击的扩散。这样，二级受害主机迅速完成了从受害者到“帮凶”的角色转换。

第八步，一级受害主机发现它作为攻击者已经成功的感染了一定数量的主机（目前已发现版本的数量为 2000）后，允许自身暴露。

第九步，随机生成 IP 作为二级受害主机的超集，对这些 IP 所在的 C 类网段的 80 口进行横向扫描（检测是否为 WWW 服务器），保存结果，获得存在 WWW 服务的主机集合（即二级受害主机集合二）；

第十步，对二级受害主机集合尝试 UNICODE 攻击（该攻击针对 Windows NT 系列系统），试图修改其主页。

用树状图，CUC 攻击的感染过程和影响规模可以被更清晰的描述出来。



（说明：1.攻击扩散时，需要通过 ftp 获得部分攻击环境基础设施；

2.攻击扩散目标随机生成，重复感染的概率较小，所以可以近似的用树性结构来描述）

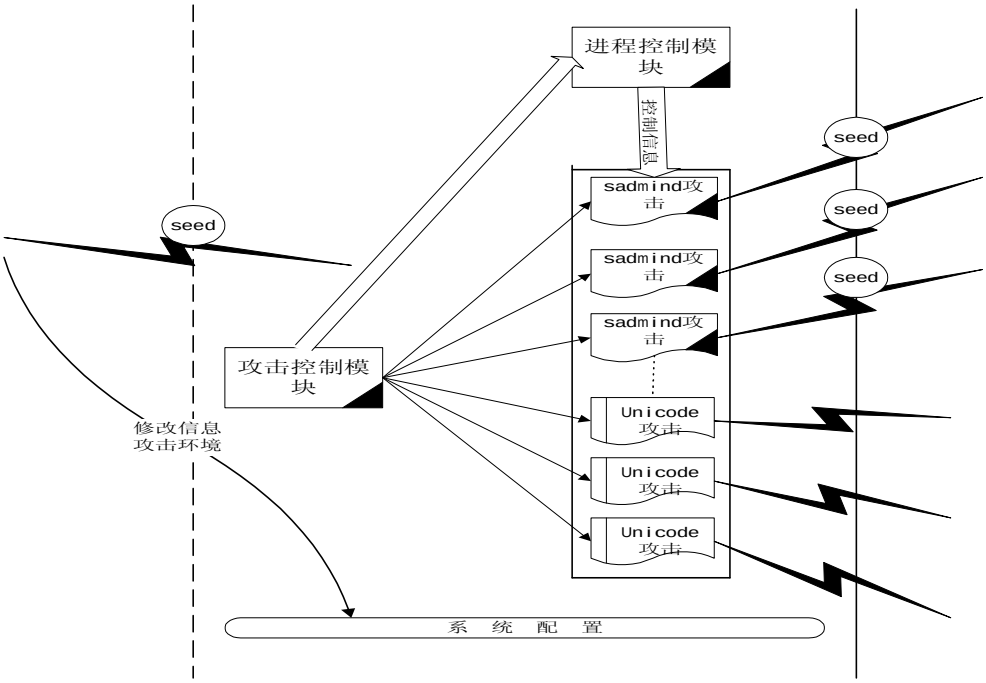
从以上的攻击行为描述可以得出，该攻击系统攻击的手段分为两类：针对 UNIX 环境的栈溢出攻击；针对 Windows NT 系列的 UNICODE 攻击。前者攻击成功后，系统被植入感染种子，能将危害扩散；而后者只会被修改主页，并不能真正被感染。因此，严格的说只有针对 UNIX 环境的攻击行为，才具有扩散性，CUC 攻击系统实际是 UNIX 蠕虫。

在目前 Internet 的规模下，随机生成的 IP 相撞概率极低。一台受害主机能感染 N 台（N<=2000）后一级受害主机，因此在该系统扩散起始的一个阶段内，感染规模呈指数级递增。在安全防范较差的网络环境下，该指数函数的底数达到10^1的数量级。与 1988 年发作的“蠕虫”相比，CUC 攻击系统缺少重复感染控制机制。实际检测中，NJCERT 发现 CUC 系统的蔓延有重复感染现象，因此可推断它的随机数生成算法的运算结果并不均匀分布在$[\ 0, 2^{32}\ ]$区间内。

2.2 CUC 攻击的结构描述

CUC 攻击系统具有一定的隐蔽性。首先，攻击程序隐藏在/dev/cuc 目录下，由于/dev 目录下存在名为 cua 的设备，因此系统管理员易认为该目录也是一个系统设备，从而使攻击系统自身得以保存；其次，CUC 攻击系统发作时，对攻击进程进行控制，使其不致占用过多的系统资源，这一点与 1988 年的“蠕虫”攻击非常相似。

下面用结构框图对 CUC 攻击系统做框架描述。



其中，各模块的功能如下：

- seed：由入侵者（原始入侵者或者前一级受害主机）植入。首先修改系统配置，在 root 的 home 目录下的.rhosts 文件中添加“＋ ＋”，使远程主机能访问该系统，并且在系统/etc/rc2.d/S71rpc 文件中添加“/bin/nohup /dev/cuc/start.sh >/dev/null 2>&1 &”，保证系统重新启动后，攻击控制模块同时被启动；其次解压缩攻击程序包，保存在/dev/cuc 目录下；再次从远地 ftp 库获得运行入侵工具的软件(perl 解释器)；最后启动攻击控制程序。
- 攻击控制模块：首先激活进程控制模块；其次采用死循环方式不断启动 5 组 sadmind 攻击和 5 组 Unicode 攻击。
- 进程控制模块：每隔 300 毫秒监听系统进程状况，当有攻击进程存活超过 300 毫秒，即将其杀死。同时统计扩散主机数量，当扩散主机超过 2000 台时，修改其主页（如果主页存在）。其中进程控制代码如下：

```
/bin/ps -ef|/bin/grep uniattack.pl > /dev/cub/tmp1
while true
do
/bin/sleep 300
/bin/ps -ef|/bin/grep uniattack.pl > /dev/cub/tmp2
/bin/awk '{print $2}' /dev/cub/tmp1 > /dev/cub/tmp3
process=`/bin/awk '{print $2}' /dev/cub/tmp2`
for p in $process;do
/bin/grep $p /dev/cub/tmp3
if [ $? = 0 ];then
/bin/kill -9 $p
fi
done
```

- sadmind 攻击模块：对 UNIX 系统实施 sadmind 攻击，并进行系统扩散。

- Unicode 攻击模块：对 Windows NT 系统实施 Unicode 攻击，修改其主页。

攻击系统的结构并不复杂，但是他行之有效的感染了相当数量的系统（NJCERT 从四月下旬到五月上旬共接到同类报告六起，受感染主机超过八台）。

3. CUC 攻击所利用的系统漏洞

3.1 sadmind 栈溢出漏洞

sadmind 是 Solaris 2.X 默认启动的 RPC 服务，该服务是分布服务的管理服务器，但是，由于其对于栈处理的不严格，导致远程攻击者可以获得系统特权用户。

通过对 CUC 攻击程序包内名为 brute 的可执行代码的跟踪执行，发现该程序发送的报文和系统调用过程与流行的 sadmind 栈溢出攻击程序完全相同。因此，可以推断出名为 brute 的可执行代码为 sadmind 栈溢出攻击 freeware 的更名版本。

3.2 Unicode 攻击漏洞

Microsoft NT4.0 和 Microsoft 2000 上的 IIS4.0 和 5.0 对扩展 Unicode “/” 和 “\” 的使用存在漏洞，攻击者利用这个漏洞，能执行远程 WWW 服务主机上的部分命令，获得系统的部分控制权，进而对网页进行修改。

在 CUC 攻击程序包内存在名为 unicodeattack.pl 的攻击程序，通过阅读源代码可以得知，CUC 攻击正式利用上述漏洞对 NT+IIS 构建的 WWW 服务器进行主页的修改。

程序摘录如下：

```
sendraw("GET/scripts/..%c0%af../winnt/system32/cmd.exe?/c+copy+\\winnt\\system32\\cmd.exe+root.exe
HTTP/1.0\r\n\r\n");

sendraw("GET/scripts/root.exe?/c+echo+^<html^>^<body+bgcolor%3Dblack^>^<br^>^<br^>^<br^>^<br^>^<br^>^
<br^>^<table+width%3D100%^>^<td^>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>fuck+USA+
Government^</font^>^<tr^>^<td^>^<p+align%3D%22center%22^>^<font+size%3D7+color%3Dred^>fuck+PoizonBOx^
<tr^>^<td^>^<p+align%3D%22center%22^>^<font+size%3D4+color%3Dred^>contact:sysadmcn\@yahoo.com.cn^</html^>>../$c/index.asp HTTP/1.0\r\n\r\n");

……
```

4. 结论

CUC 攻击系统主要尝试针对不同操作系统类型的两种攻击方式（Unix-sadmind;Windows NT-unicode），对位于 Internet 的主机进行攻击。对于 Unix 主机，该系统具有扩散性，表现出“蠕虫”的特点；系统长时间潜伏在主机中，为了增加自身隐蔽性，它对系统派生的进程进行了严格的控制。与 1988 年出现的蠕虫相比，CUC 网络攻击系统采用了不同的进程控制方式，但是缺乏明显的感染控制机制。

CUC 攻击近期造成一定的影响。目前网络攻击行为趋向自动化、规模化，大规模的网络进攻例如“蠕虫”、DDoS 攻击越来越常见，因此网络应急响应、单位之间的协作与协同显得尤其重要。

5. 参考文献（略）